TEC-0077

Representation, Modeling and Recognition of Outdoor Scenes

Martin A. Fischler
Robert C. Bolles

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025-3493

January 1996

U.S. Army Corps of Engineers
Topographic Engineering Center
7701 Telegraph Road
Alexandria, Virginia 22315-3864

US Army Corps of Engineers
Topographic Engineering Center

T

E

C

Destroy this report when no longer needed.
Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The citation in this report of trade names of commercially available products does not constitute official endorsement or approval of the use of such products.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1996	Third Annual April 1994 - April 1995

4. TITLE AND SUBTITLE

Representation, Modeling and Recognition of Outdoor Scenes

5. FUNDING NUMBERS

DACA76-92-C-0008

6. AUTHOR(S)

Martin A. Fischler
Robert C. Bolles

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025-3493

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Topographic Engineering Center
7701 Telegraph Road
Alexandria, VA 22315-3864

19. SPONSORING / MONITORING AGENCY REPORT NUMBER

TEC-0077

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The goal of this project is to advance the state-of-the-art in scene interpretation for autonomous systems that operate in natural terrain. In particular, techniques are being developed for representing knowledge about complex cultural and natural environments so that a computer vision system can successfully plan, navigate, recognize and manipulate objects, and answer questions or make decisions relevent to this knowlegde. The results to date include the development of new representations for rapidly modeling terrain from multiple images -- including 3-D hexagonal surface meshes and "particle-based" 3-D modeling primitives that allow a robotic system to incrementally build a complete geometric model of its environment. We have also extended and improved our set of object recognition primitives to allow reliable labeling of such scene attributes and components as color, texture, shadows, and a variety of linear structures (skyline, ridgelines, roads, etc.). The most recent results are detailed in four published papers included as appendices to this report.

14. SUBJECT TERMS

Machine Vision, Automated Scene Analysis, Object Recognition, Terrain Modeling

15. NUMBER OF PAGES

117

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

TABLE OF CONTENTS

PREFACE

This research is sponsored by the Advanced Research Projects Agency (ARPA) and monitored by the U.S. Army Topographic Engineering Center (TEC) under contract DACA76-92-C-0008, titled "Representation, Modeling and Recognition of Outdoor Scenes". The ARPA Program Manager is **Dr. Thomas M. Strat,** and the TEC Contracting Officer's Representative is **Ms. Lauretta Williams.**

1.0 OBJECTIVE

The primary goal of this project is to advance the state-of-the-art in scene interpretation for autonomous systems that operate in natural terrain. In particular, techniques are being developed for representing knowledge about complex cultural and natural environments so that a computer vision system can successfully plan, navigate, recognize, and manipulate objects, and answer questions or make decisions relevant to this knowledge.

2.0 APPROACH

This work integrates our continuing advances in four separate technologies to achieve the goal of providing a foundation for the design of highly competent machine vision systems capable of autonomous operation in the outdoor world:

- Stored knowledge (such as map data and object models) is used to overcome inherent weaknesses in the best "self-contained" image-analysis algorithms. This approach is reflected in the prior SRI development of the CONDOR system that relies on context, function, and purpose, as well as visually observed geometric shape, to recognize scene objects.

- Significant progress has been made in developing compact and expressive representations for modeling, and ultimately, recognizing objects encountered in the natural world. Computational efficiency, thus, real-time performance, is critically dependent on using effective representations for both models and sensed data.

- Global optimization techniques are being developed that require reasonable amounts of computation, but produce results not obtainable by local analysis methods. This work has been applied to building volumetric models of objects detected in range data and stereo pairs, as well as for delineation, partitioning, and feature extraction in single images.

- Techniques are being developed that are able to simultaneously, or incrementally, exploit multiple views of a scene in compiling a complete scene model. The SRI-developed Epipolar plane image analysis technique is one example of how motion sequences can be used to construct a geometric scene model that is superior to a sequence of independent stereo reconstructions.

3.0 PROGRESS

As indicated, both theoretical and practical progress has been made in integrating semantic and geometric information about a scene from stored knowledge, multiple views, and image sequences. The results, to date, are centered on the development of representations and associated methods for rapidly modeling natural terrain (at a level of organization higher than that of the conventional dense array of depths), and on methods for recognizing important classes of natural and man-made objects -- especially roads, trees, rocks, and terrain features.

We are also developing metrics for objective evaluation of scientific progress in the domain of natural object recognition (we have taken a lead position in ARPA's benchmarking efforts in this area). Fourteen papers have been published describing this work, and some of the algorithmic techniques developed in this program are currently being integrated into a commercial cartographic modeling system.

4.0 SUMMARY OF RECENT ACCOMPLISHMENTS

4.1 Recovery of Scene Geometric from Multiple Views

The development of an approach for integration of information acquired from multiple views of a scene into a description of scene geometry continued. The approach uses a new class of geometric primitives that permits simple expression of known constraints and observed data, and also permits the use of practical optimization-based solution techniques. This work will provide an effective way of allowing a robotic system to incrementally build a progressively more accurate and complete model of the environment in which it is operating.

Earlier work on this task involved the use of 3D hexagonal meshes. Our most recent work involves recovering 3D surfaces of arbitrary topology from multiple stereo views employing a new "particle-based" method. We start with several stereo-pairs of a scene and use a conventional stereo-correlation algorithm to compute a disparity map for each pair. Given camera models, the maps can be turned into "clouds" of 3D points. These raw points are typically very noisy and a non-negligible fraction of them may be in error. To overcome this problem, a three-step approach was devised:

1. Robust fitting of local surfaces to the points: Divide the 3D space into buckets and fit a quadric patch in each bucket that contains at least a minimum number of points. This first step eliminates the isolated errors and reduces the large number of 3D points to a smaller and more manageable set of patches defined by their center of gravity and normal.

2. Global optimization of the patches: The positions of these patches can then be optimized by minimizing an image-based objective function and interacting with each other through forces that tend to align those that appear to belong to the same surface.

3. Grouping of the patches into global surfaces: Patches that appear to belong to the same global surface---that is, patches that are close and whose normals are consistent---are grouped into global surfaces that can then be triangulated.

These three steps have been implemented and tested on both outdoor and indoor scenes. For example, ground-level outdoor scenes containing rocks and trees, and scenes containing complex objects---such as a complete human head. The method successfully recovers the main surfaces in the scene and rejects erroneous points from the correlation-based stereo data that we use as input to our system.

Ten papers describing the work in this task area have been accepted for journal publication and conference presentation. Enclosed are reprints of two of the published papers that provide details about the implementation of both 3D meshes and particles.

4.2 Natural Object Recognition

The problem of automatically recognizing objects appearing in images of the outdoor world has proven to be extremely difficult because of the lack of explicit shape models. Most computer-based recognition techniques rely on explicit knowledge of shape, however rocks, trees, and other natural objects cannot be successfully described in this way; even such generic man-made objects as roads, bridges, and buildings are more likely to satisfy functional constraints rather than being exemplars of some geometric

blueprint. It is necessary to replace explicit shape with a more general way of describing natural objects and complex man-made structures. What is required are a few techniques that can reliably organize the pixel-level image data as a basis for higher-level analysis. Finding the appropriate combination of low-level data description and associated extraction techniques is a key problem in machine vision, and one of the primary concerns in this project. Two of the techniques that have emerged from the work in this area meet the criteria of generality and robustness. The first is a generic way to find candidate line structure in an image; some of this work is described below and also in a paper published in the 1994 IU workshop proceedings (reprint in Appendix A). The second is a way to organize such data into perceptually coherent and semantically meaningful units. In another paper (reprint in Appendix B), we describe our progress in the design of a partitioning technique that is extremely robust in accomplishing the perceptual organization task and also describes how these two techniques can be applied to the problem of road delineation in aerial images. Our most recent work focuses on the development of additional algorithms for the recognition of a wide range of natural objects of importance to robotic navigation and outdoor scene modeling. Some of this work is described below, and a paper is in preparation.

4.3 Obstacle Detection for Robotic Navigation

To be employed in practical outdoor robotic systems of military interest, our scene interpretation techniques must be fast, reliable, and have modest resource requirements; on the other hand, full object recognition is often not required. We have adapted some of our recognition techniques to satisfy the above requirements in detecting raised-object and depression-type obstacles to cross-country robotic vehicle navigation. For these types of obstacles, a significant part of the localization problem can be captured in a differential description of the discontinuities in a stereo disparity image. A method that locates these discontinuities quickly, and measures the differential properties of the disparity surface on either side of each such discontinuity has been developed. This is done with a family of recursive filters that form one-sided estimators of differential properties. Points of maximal difference between opposing operators are taken as discontinuities and the estimates around these points are used to measure both the disparity difference and the gradients around the discontinuity. These values are then translated to depths, depth discontinuities, and surface slopes. To reason in this way, using a disparity image, we must have a reliable, dense image. However, stereo mismatches produce unreliable disparities, extraneous discontinuities, and (even if eliminated) gaps in the disparity image. Bad matches are eliminated by using only those stereo matches that are "best matches" in both the left/right and right/left directions. Once this is done, we take advantage of the forward-looking ground-based nature of these images and eliminate all matches that do not belong to a monotonically decreasing sequence of disparities from the bottom to the top of the image. (This technique assumes that there are no observable overhangs, a constraint that is reasonable in many environments and viewing situations.) Once these problem disparities are eliminated, the image is filled in by interpolating values from neighboring regions. The resulting disparity image is reliable, dense, and locally smooth. It is, thus, well suited for the analysis described.

We also continue work to build simplified geometric descriptions of the terrain using triangulated meshes. A mesh is first overlayed on the disparity image such that no facets cross discontinuities; then, it is reduced in complexity so as to minimize the number of triangles in the mesh. The technique developed for this is similar to "mesh decimation" used in graphics, but constant reference is made to the initial disparity image, thus ensuring that the final mesh is still an accurate representation of the disparity image. We have been able to achieve data reductions of over 98% of the data in the original mesh.

When this mesh is transformed into real-world coordinates, we have a simplified, still accurate, triangulated description of the visible surfaces.

4.4 Line Drawing Interpretation from Man Machine Communication

Earlier in this project, we made a significant new advance in the long-standing problem of duplicating human performance in recovering 3D models of terrain and man-made objects from qualitative and imprecise line drawings (e.g., of terrain elevations as in an approximate and uncalibrated contour map, or building edges, as in a single approximate projection of the corresponding wire-frame). This work can greatly simplify communication problems between man and machine in such applications as robotic mission planning, and in construction of databases for use in robotic navigation. A paper describing this work has been published.[1] Ongoing work has led to additional (new) results of both theoretical and practical importance; these new results, still being further developed and evaluated, will be described in a later report.

5.0 DETAILED DISCUSSION OF WORK ON GEOMETRIC RECOVERY

The recovery of surface shape from image cues, the so-called "shape from X" problem, has been the focus of a major effort in the computer vision community; no single source of information "X," be it stereo, shading, texture, geometric constraints or any other, has proved to be sufficient across a reasonable sampling of images. To get good reconstructions of a surface, it is necessary to use as many different kinds of cues with as many views of the surface as possible. In this effort, we have devised a working framework for surface reconstruction that combines diverse image cues, such as stereo and shape-from-shading, obtained from multiple images whose vantage points may be very different.

The task is difficult because image analysis techniques typically provide incomplete, and sometimes erroneous, information about the location of 3D points in space. Grouping these points into meaningful surfaces involves solving a problem akin to the notoriously hard segmentation problem. As a result, a majority of recent computer vision approaches to surface reconstruction rely on much cleaner sources of 3D data -- such as laser range maps or medical volumetric data -- as their input. Many of these approaches also assume that there is one, and only one, object of interest whose topology is known in advance so that a particular model -- be it rigid or deformable -- can be fit to the data.

More specifically, to recover surfaces from images, one must contend with the following difficulties:

- Real-world scenes often contain several objects whose topology may not be known in advance: some surfaces are best modeled as sheets while others are topological spheres or contain holes. One cannot typically assume that there is only one object and one surface of interest or expect to be able to easily cluster the 3D points derived from stereo into semantically meaningful groups.

- The 2-1/2D assumption that most traditional interpolation schemes make is no longer valid when reconstructing complex 3D scenes from arbitrary numbers of images and arbitrary viewpoints. Surfaces often overlap and may be visible in one view, but not another.

---

[1] An Optimization-Based Approach to the Interpretation of Single Line Drawings as 3D Wire Frames, IJCV 9(2):113-136, Nov. 1992

- The 3D points derived from range maps--computed using passive or active methods--form an irregular sampling of the measured surfaces. In addition, small errors in disparity can result in large errors in world position.

- Even the best vision algorithms make occasional blunders that must be identified and eliminated. Furthermore, the corresponding erroneous 3D points are often correlated with one another so that they cannot be eliminated by robust estimation alone.

To overcome these problems, we need a surface representation that can be used to generate images of the surface from arbitrary viewpoints, taking into account self-occlusion, self-shadowing, and other viewpoint-dependent effects. Clearly, a single image-centered representation is inadequate for this purpose. Instead, object-centered surface representations are required. Here, we advocate the use of two such representations that have proved effective under different circumstances:

- Triangulated Meshes: A regular 3D triangulation is an example of a surface representation that meets the criteria stated above. Our approach to recovering surface shape and reflectance properties from multiple images is to deform a 3D representation of the surface so as to minimize an objective function. The free variables of this objective function are the coordinates of the vertices of the triangulation, and the process is started with an initial surface estimate. We have successfully used these meshes to model surfaces whose topology is known a priori, such as human faces or relatively low-resolution terrain.

- Sets of 3D Particles: Real-world scenes often contain several objects whose topology may not be known in advance -- some surfaces are best modeled as sheets, while others are topological spheres or contain holes. One cannot typically assume that there is only one object and one surface of interest. To deal with these complex issues, we replace the triangulations by a set of connected surface patches or "oriented particles." These particles are instantiated by fitting local surfaces to traditional stereo data, and their positions are refined by minimizing an objective function analogous to the one used for triangulations.

Both of these techniques are described in detail in the enclosed two reprints (Appendices C and D).

We view the contribution of this work as providing both the framework that allows us to combine diverse sources of information in a unified and computationally effective manner, and the specific details of how these diverse sources of information are derived from the images.

6.0 DETAILED DISCUSSION OF WORK ON NATURAL OBJECT RECOGNITION

In this effort we are developing and evaluating two distinct approaches to natural object recognition. The first is based primarily on geometric information about the scene as obtained from stereo or some other form of range sensing. The second approach is based on using information about illumination, shadows, shape, color, and texture as extracted from a single color image of the scene (or possibly from multiple images -- but not requiring the explicit availability of range data). The first steps of merging this work in these two approaches has been taken.

6.1 Geometric Techniques for Recognizing and Modeling Terrain Features

Because of their importance to unmanned ground vehicle (UGV) navigation, techniques are being developed for the recognition and delineation of rocks and other relatively small objects and terrain features. We intend to make these techniques sufficiently general so as to allow their use in finding a wide range of compact objects protruding above the ground surface.

In one part of this investigation, we are concerned with locating rock-like objects and obstacles within the operating environment of a forward-looking pair of stereo sensors. Here we adopt a very simple geometric model of these obstacles in order to satisfy the requirements for high-speed detection and limited computing power needed for practical UGV applications.

We begin by assuming that these obstacles are distributed sparsely enough so that we may examine each in relative isolation and locate them by using only simple local patterns in the disparity image(s). When the obstacles are spatially separated, the detection patterns depend primarily on the conjunction of disparity discontinuities (e.g., occlusion edges) and stereo shadows (regions visible from only one camera). We are currently conducting experiments to determine whether or not this approach is robust in the presence of multiple objects overlapping along a line-of-sight. A key feature of the local pattern used to locate these obstacles is the "top" of the object, a near-horizontal spatial discontinuity. This feature is flanked by either a stereo shadow region or a high disparity gradient. If both L-R and R-L disparity images are made available, then mirror images of this pattern may be sought in the two disparity images. The coordinated detection of these features locates the presence and extent of such obstacles.

For both rock-and ditch-like obstacles, a significant part of the localization problem can be captured in a differential description of the discontinuities in the stereo disparity image. We have a method that locates these discontinuities quickly, and measures the differential properties of the disparity surface on either side of each such discontinuity. The discontinuities are found with a conventional Canny/Deriche filter and the differential properties are computed using a set of least-squares-based one-sided estimators oriented perpendicular to the extracted edges.

In the simplest application (to obstacle detection) of the above approach, varying the thresholds throughout the image, dependent on the stereo geometry, allows the detection of obstacles based solely on size. This type of analysis depends critically on the parameters of the low-level stereo fusion used. The resolution determines the lower limit on the size of obstacles detectable. The amount of smoothing used to condition the results also affects detection, especially if the method used obscures discontinuities. In general, however, without additional texture or chromatic input, it is difficult to distinguish between obstacles (e.g., rocks) and nonobstacles (e.g., clumps of grass). In the work discussed below, we respond to the need to distinguish actual hazards, such as rocks, from benign, but similarly shaped objects, such as bushes.

6.2 Semantic Scene Description

Our goal in this subtask is to be able to recognize natural objects and terrain features in the context of creating an overall scene sketch. We are not only interested in recognizing (and delineating) isolated objects, but wish to describe and exploit their interrelationships. Objects of interest include rocks, trees, brush, grass, water, snow, dirt, sky, ridgelines, holes/ditches, roads, paths, fences, poles, cliffs, ground plane, and shadows.

A key problem is the obvious necessity to replace reliance on (generally unavailable) explicit shape with more general ways of recognizing and describing natural objects and complex man-made structures. Our approach was to first select (or define) a smaller set of primitive (but pervasive) features that can reliably be extracted from most images of natural scenes. These primitives (currently consisting of color, texture, shadows, depth, surface orientation, and linear structures) are combined to identify clear instances of the natural objects of interest using a "production rule" type paradigm[2], and then, using these recognized objects as exemplars, we invoke a nearest-neighbor statistical classifier to label other, possibly less obvious, instances of the objects we are looking for.

Thus, for example, we have implemented a color-based classification algorithm using classical feature-space partitioning techniques and run a significant number of experiments on outdoor images from a variety of sources. We have been successful in distinguishing certain categories of objects, such as between vegetation and rocks (or other nonliving terrain features). While not sufficient to produce a complete semantic labeling of a scene at a detailed level of recognition, this approach appears able to complement our geometric recognition techniques and allow us to successfully deal with some of the critical open problems -- such as distinguishing between real obstacles (e.g., rocks) and navigable areas (e.g., grass or brush-covered flat terrain). A paper describing this work is currently in preparation.

Techniques to recognize the more important and prominent extended linear terrain features and navigation obstacles are being developed; these include the skyline, ridgelines, and the leading edge of drop-offs and ditches. These features appear as significant linear discontinuities in the disparity image, and as intensity discontinuities in a color or grayscale image. By first locating depth/disparity and intensity or color discontinuities, and then linking these local features, we can find and label both small compact objects and the major extended linear scene features. For example, by first addressing the simpler problem of delineating the skyline in a color image, we are then able to choose a region above the skyline as an exemplar of "sky" for use by the color classifier. Texture, shadows, and shape will allow us to find a few obvious instances of vegetation and rocks as additional exemplars for the color classifier. We can now label most of the scene using the color classifier, and then (at least partially) check the result for semantic consistency: The pixels labeled sky by the color classifier should all exit above the skyline found by the linear delineation process; if the skyline is interrupted by a nearby (as measured by our depth measuring techniques), thin, raised object, the object should be labeled as a tree or a pole; and a relatively horizontal/flat region, depending on its color, should be either grass, dirt, water, or rock, and so forth.

The key ideas underlying this work are:

- Models described by objective functions referenced to some appropriate representation, and feature extraction accomplished by finding image structures for which the relevant objective function is optimized.

- Recognition-technique selection and corresponding parameter settings based on context.

---

2 Strat and Fischler, "Context-Based Vision," IEEE PAMI, Oct. 1991

- Selection and intense development to produce a few highly refined and reliable "core" techniques as the base for implementing a much broader class of feature recognition/extraction methods.

In an enclosed paper (Appendix A), we describe the most recent work on the detection and extraction of linear features in imaged data -- one of the core techniques. Following the paradigm outlined above, we use the minimum spanning tree, and a new "network" structure we devised, as the primary representations. Semantic constraints control the tree/network construction, and thus, establish the universe of possible paths (both in our data structures and in the image being analyzed). We define the characteristics of the linear structures we are looking for as attributes of the branches in the tree/network, and provide computationally effective methods for finding paths that maximize scores for the desired attributes. Filtering techniques, parameterized by context evaluation procedures (or externally provided information) operate at a number of decision points in the optimization process, and in final acceptance of the selected path(s). We have implemented specialized versions of the generic delineation technique to recognize various types of extended terrain features and navigation obstacles including the skyline, ridgelines, trees, roads, and paths.

As an additional part of this effort to construct an overall scene sketch, we are developing a stereo interpretation system for visual navigation that produces a sparse triangular-mesh representation of the ground geometry, and that is augmented with semantic labels for obstacle-like structures.

We begin by edge-processing the disparity image using two thresholds. The first is a constant that covers uncertainty in the stereo algorithm that generates the disparities. The second varies with the disparity so that only discontinuities corresponding to significant real-world discontinuities are retained. A dense triangular mesh is then constructed over the disparity image so that no mesh elements cross the detected discontinuities. Finally, a fast decimation procedure is applied to the mesh to minimize the number of triangles needed to represent the surface. The resulting mesh is guaranteed to not deviate from the initial disparity data by more than a given threshold value (usually the same one used as the constant edge threshold), and has mesh boundaries only at internal discontinuities and data boundaries (which can be distinctly labeled). We have been able to achieve data reductions of over 98% of the data in the original mesh. When this mesh is transformed into real-world coordinates, we have an accurate, triangulated description of the visible surfaces.

The mesh structure can now be efficiently labeled to represent geometrically and semantically significant properties of the image and environment. For example, we need only project the vertices of the mesh into three-dimensional space in order to project the entire mesh. Once this is done, we can relabel the discontinuous boundary edges in the mesh, depending on whether the leading surface is tangent to the line of sight or not (a limb edge versus a surface discontinuity).

Finally, the shape and extent of the discontinuities in the image are used to identify potential obstacles. Localized depth discontinuities are associated with the tops and sides of objects such as rocks, while extended discontinuities in the ground surface are associated with ditches.

BIBLIOGRAPHY
(Publications describing work performed on this project)

Y.G. Leclerc and M.A. Fischler "An Optimization Based Approach to the Interpretation of Single Line Drawings as 3D Wire Frames," IJCV 9(2):113-136, Nov. 1992.

P. Fua and Y.G. Leclerc, "Combining Stereo, Shading, and Geometric Constraints for Surface Reconstruction from Multiple Views," Technical Conference on Geometric Methods in Computer Vision II of SPIE Symposium, San Diego, CA, Jul. 1993.

P. Fua and Y.G. Leclerc, "Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading," IJCV, 1994.

P. Fua and Y.G. Leclerc, "Using 3-Dimensional Meshes to Combine Image-Based and Geometry-Based Constraints," ECCV, Stockholm, Sweden, May 1994.

P. Fua and Y.G. Leclerc, "Registration without Correspondences," CVPR, Seattle, WA, pp. 121-128, Jun. 1994.

P. Fua and Y.G. Leclerc, "A Unified Framework to Recover 3D Surfaces by Combining Image-Based and Externally-Supplied Constraints," Procedures, ARPA Image Understanding Workshop, Monterey, CA, Nov. 1994.

P. Fua and Y.G. Leclerc, "Image Registration without Explicit Point Correspondences," Procedures, ARPA Image Understanding Workshop, Monterey, CA, Nov. 1994.

P. Fua, "Surface Reconstruction Using 3D Meshes and Particle Systems," Third International Workshop on High Precision Navigation, Stuttgart, Germany, Apr. 1995.

P. Fua, "Reconstructing Complex Surfaces from Multiple Stereo Views," (submitted to ICCV), Boston, MA, Jun. 1995.

M.A. Fischler and H.C. Wolf, "Saliency Detection and Partitioning Planar Curves," Procedures, Image Understanding Workshop, Washington DC., pp. 917-931, Apr. 1993.

M.A. Fischler and H.C. Wolf, "Locating Perceptually Salient Points on Planar Curves," IEEE-PAMI, vol. 16(2):1-17, Feb. 1994.

M.A. Fischler, "The Perception of Linear Structure: A Generic Linker," Procedures, ARPA Image Understanding Workshop, Monterey, CA, Nov. 1994.

W. Neuenschwander, P. Fua, G. Szekely, and O. Kubler, "Initializing Snakes," CVPR, Seattle, WA, Jun. 1994.

W. Neuenschwander, P. Fua, G. Szekely, and O. Kubler, "Using Boundary Conditions to Improve Snake Convergence," ICPR, Jerusalem, Israel, Oct. 1994.

APPENDICES

M.A. Fischler and H.C. Wolf, "Locating Perceptually Salient Points on Planar Curves," IEEE-PAMI, vol. 16(2):1-17, Feb. 1994.

M.A. Fischler, "The Perception of Linear Structure: A Generic Linker," Procedures, ARPA Image Understanding Workshop, Monterey, CA, Nov. 1994.

P. Fua and Y.G. Leclerc, "Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading," IJCV, 1994.

P. Fua, "Reconstructing Complex Surfaces from Multiple Stereo Views," (submitted to ICCV), Boston, MA, Jun. 1995.

Appendix A

"Locating Perceptually Salient Points on Planar Curves"

IEEE-PAMI, vol. 16(2):1-17, February 1994

Martin A. Fischler and Helen C. Wolf

Locating Perceptually Salient Points on Planar Curves

Martin A. Fischler, *Member, IEEE*, and Helen C. Wolf

Abstract—**This paper describes the underlying ideas and algorithmic details of a computer program that performs at a human level of competence for a significant subset of the *curve partitioning* task. It extends and rounds out the technique and philosophical approach originally presented in a 1986 paper by Fischler and Bolles. In particular, it provides a unified strategy for selecting and dealing with interactions between salient points, even when these points are salient at different scales of resolution. Experimental results are presented involving on the order of 1000 real and synthetically generated images.**

Index Terms—**Computer vision, salient points, critical points, curve partitioning, curve segmentation, curve description.**

I. Introduction

A CRITICAL problem in machine vision is how to break up (partition) the perceived world into coherent or meaningful parts prior to knowing the identity of these parts. Almost all current machine vision paradigms require some form of partitioning as an early simplification step to avoid having to resolve a combinatorially large number of alternatives in the subsequent analysis process. Given this critical role for partitioning as a functional requirement of a complete vision system, it is a major challenge to find some significant subset of the partitioning problem for which an algorithmic procedure can duplicate normal human performance. This paper describes the underlying ideas and algorithmic details of a computer program that performs at a human level of competence for a significant subset of the *curve partitioning* task. It extends and rounds out the technique and philosophical approach originally presented in a 1986 paper by Fischler and Bolles [6]. For example, it provides a unified strategy for resolving conflicts in selecting among neighboring potential partition points that may be salient at different scales of resolution.

While our focus in this paper is on curve partitioning in a generalized setting (the curves in our experiments are mostly without semantic meaning), and where the criterion for success is duplicating normal human performance, finding salient points on image curves (potential partition points) plays a critical role in both two- and three-dimensional (2-D and 3-D) object recognition, in curve approximation, in tracking moving objects, and in many other tasks in machine vision.

In many approaches to 2-D object recognition, objects are represented by their boundaries, and the recognition techniques depend (directly or indirectly) on locating distinguished points along the boundary; typically these distinguished points are discontinuities or extrema of local curvature (sometimes called corner points) and inflection points (e.g., [14]). In 3-D recognition, partitioning is typically one of the first analysis steps—especially when objects can occlude each other. Hoffman and Richards [10] argue that when 3-D parts are joined to create complex objects, concavities will generally be observed in their silhouettes, and that segmentation of image contours at concavities (the maxima of negative curvature along the contours) is a good strategy to decompose (even unmodeled) objects into their "natural parts."

In cartography, computer graphics, and scene anaysis, it is often desirable to partition an extended boundary or a contour into a sequence of simply represented primitives (e.g., straight line segments or polynomial curves of some higher degree) to simplify subsequent analysis and to minimize storage requirements (e.g., [21]). We present some examples of the use of our algorithm for this purpose (see Figs. 9 and 10 and the brief discussion on this topic in Section VII).

Corners on the contours of imaged objects are often used as features for tracking the motion of these objects and for computing optical flow [5], [16], [13].

In our current work concerned with delineating linear structures in aerial images, the technique presented in this paper was an essential component of the system (briefly described in Appendix E) that produced the results displayed in Fig. 17.

II. Problem Statement

In its most general sense, partitioning involves assigning, to every element of a given object set, a label from a given label set. For our purposes in this paper, the object set is the set of points along a curve (or contour segment) lying in a prescribed region of a 2-D plane. Although we deal with cases where the points in the object set do not form a continuous digital curve, in most of our exposition in this paper we will assume that the curves are continuous[1] and nonintersecting. Our label set is binary, points will be called either significant (critical) or nonsignificant, for some specified purpose. In [6] it is demonstrated (or at least argued) that perceptual partitioning is *not* independent of some assumed task or purpose. In this paper we focus on one of the three tasks discussed in the above reference: selecting a small number of points (called *critpts*) along a curve segment that could be used as the basis for *recon-*

Manuscript received August 11, 1992; revised February 3, 1993. This work was supported by the Advanced Research Projects Agency. Recommended for acceptance by Associate Editor E. Hildreth.

The authors are with SRI International, Menlo Park, CA, 94025.

IEEE Log Number 9214424.

[1] Each point of the nonbranching 1-pixel-wide curve, with coordinates (x, y), has one or more neighbors with x-coordinates in the set $(x+1, x, x-1)$ and y-coordinates in the set $(y+1, y, y-1)$.

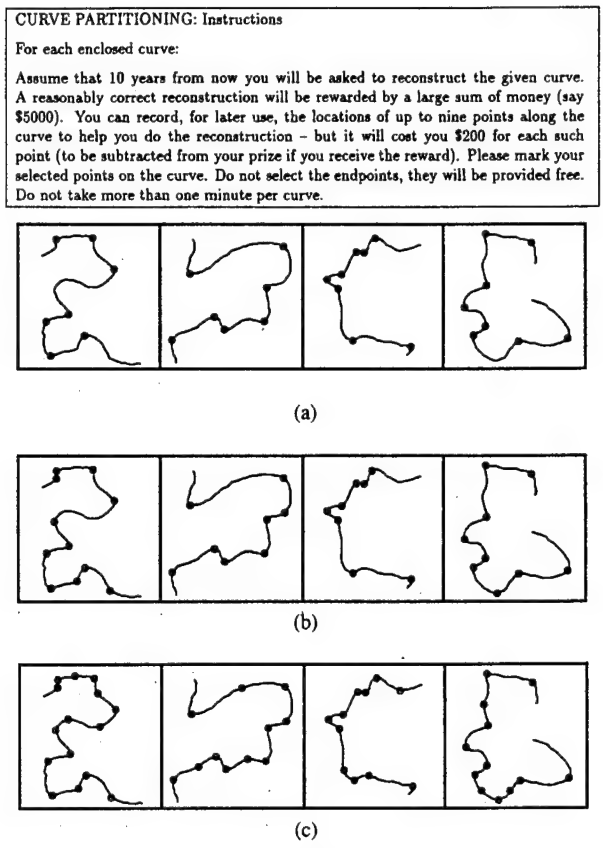


Fig. 1. Comparison of human and SSS algorithm performance in the curve partitioning task. (Each of the curves used in the experiments with human subjects was contained in a square that was 1.5 in on a side.) (a) Points chosen by 9 of 11 test subjects. (b) Critical points found by the SSS algorithm. (c) Points chosen by at least 1 of 11 test subjects.

structing the curve at some future time. Fig. 1 shows the specific instructions and curves used in one set of relevant experiments involving human subjects; this figure also shows the critpts that were selected by the subjects, and the comparable results produced by our algorithm (called the saliency selection system, or SSS, and described in detail in Appendix B).

In order to separate the generic partitioning criteria used by human subjects from criteria based on their past experience, such as when the subject is able to assign a name to the curve (e.g., the curve looks like the letter "s"), we used "random" curve segments for our experiments; the technique employed to generate the segments is described in Appendix A. We also wanted to avoid having to deal with the recognition of global features (e.g., symmetry or repeated structure, or even straight lines and analytic curves) as a condition for making critpt selections; avoiding this problem is justified if we are correct in our belief that local and global analysis are accomplished by separate mechanisms. In order to deal with global features, the complexity of any solution would be expanded enormously since a whole new vocabulary of such features and their representations would have to be implemented. The generation and use of random curves took care of this problem also (i.e., it is highly unlikely that symmetries or repeated structure would ever be generated by our random process).

III. Relevance, Prior Work, and Critical Issues

The partitioning problem has been a subject of intense investigation since the earliest work began in machine vision. It has been widely assumed that in order to reduce the combinatorics of scene analysis to a manageable level, it is necessary to decompose images into their meaningful component parts as one of the first steps in the analysis process. The difficulty arises from the need to partition the image into parts before we know the identity of those parts. The underlying assumption then is that there are generic criteria, independent of the goal of the analysis, that if discovered could be used to obtain useful (or, at least, intuitively acceptable) partitioning; additional problem dependent criteria could be always added to produce a more relevant result for some particular purpose.

The partitioning problem becomes progressively more difficult as we increase the number of dimensions in which we are working; in this paper we address only the 1.5-D problem of partitioning planar curves. A specific criterion that can form the basis of such partitioning was originally proposed by Attneave [1]—points at which the curve bends most sharply are good partition points.[2] This idea has been the starting point for most of the subsequent efforts in curve partitioning, but attempts to convert this abstract concept into a computationally executable procedure, one that gives intuitively acceptable results, have met with limited success.[3] [11], [14], [15], [19], [21], and [23] are representative of work in this area.[4]

The main problems we must solve are as follows:

1) A way of assigning a *measure (or degree) of saliency/criticality*[5] to each point on a curve. Most investigators have equated sharp bending of a curve with the mathematical concept of curvature (Appendix C), but curvature is not well defined for a finite sequence of points (which is how our sensor acquired curves are generally represented). Further, it is not obvious that the mathematical definition of curvature is the best computational approximation to the human criteria for criticality. In Fischler and Bolles [6], bending is interpreted as deviation from straightness—it is closely related to proposed approximations to mathematical curvature, as illustrated in Figs. 2 and 3, but has a number of advantages: It is an easily measured quantity, even for digital curves (i.e., sequences of coordinate pairs), and as discussed in the next section, its local extrema are in better accord with human preference

[2] Hoffman and Richards [10] give convincing evidence that we should distinguish between positive and negative curvature maxima. That is, on closed curves, extreme points of negative curvature—associated with object concavities—have greater utility as partition points than positive curvature maxima, but the positive maxima (and inflection points) play an important role in describing the individual segments.

[3] As noted below, most of the work on the curve partitioning problem, especially recent work, has *not* been concerned with duplicating generic human performance but rather with performing specific visual tasks having different criteria for success.

[4] The approach taken by Wuescher and Boyer is distinct in that they first extract contour segments of approximately constant curvature and then infer the location of partition points as a secondary operation.

[5] We will use the terms *saliency* and *criticality* somewhat interchangeably in this paper. However, saliency can be considered to be the generic subset of points that are critical for some partitioning task.

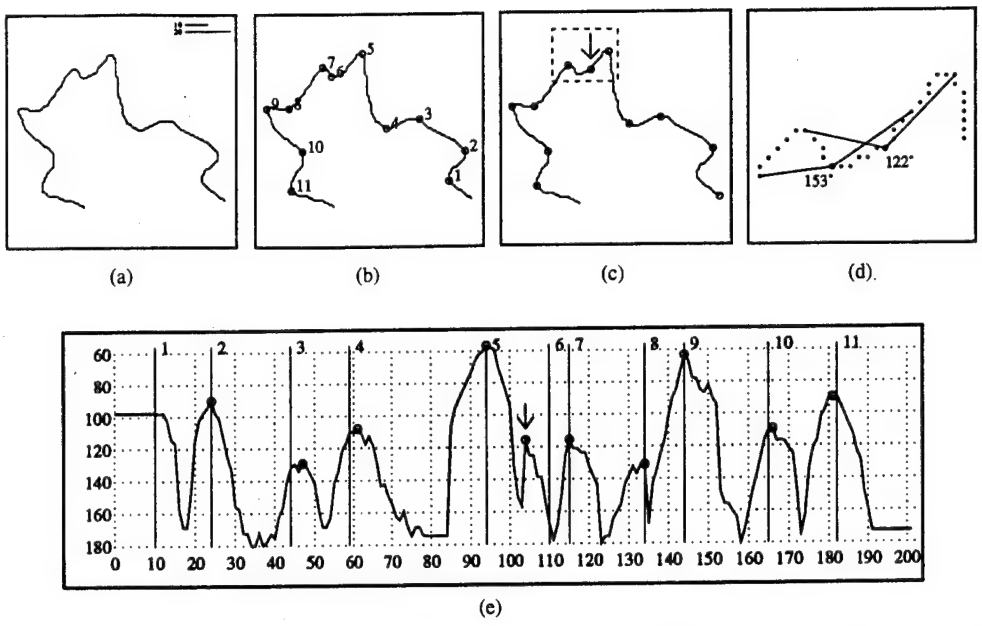


Fig. 2. Comparison of SSS and R/J-curvature evaluated on test curve 189. (a) Test curve 189. (b) SSS selected critpts. (c) R/J-curvature selected critpts. (d) Anomalous area (magnified). (e) Plot of R/J-curvature along test curve. Abscissa = sequence number of point on curve. Ordinate = angle (in degrees) computed at point. (Angle-arms are 10 units each for R/J-C; standard stick lengths of 10 and 20 units are employed by SSS.)

The continuous curve in (e) represents R/J-curvature along the test curve shown in (a). The vertical lines in (e) mark the sequentially numbered critpts selected by SSS as shown in (b). The critpts corresponding to the extreme values of R/J-curvature shown in (c) are marked as circles in (e) The arrow in (c), and in the corresponding location in (e), illustrates an anomalous selection using R/J-curvature. (d) shows the computed values of R/J-curvature, 153°, at the preferred location and 122° at the location of the anomalous selection.

(choices based on approximations to the definition of mathematical curvature occasionally include anomalous points, as shown in the examples of Figs. 2 and 3).

2) A way of adjusting the criticality of a given curve-point to take into account its interactions with its neighbors; i.e., *local context*. It is obvious that human subjects will often avoid assigning a critpt label to both members of a pair of points, even when both points have high (independent) criticality values, if the points are close neighbors along the curve. The basic approach of local nonmaximum suppression is not sufficient, in itself, to duplicate human performance.
3) A way of dealing with the interactions between critpts that are significant at *different scales of resolution*. If a human subject looks through a fixed sized window at the same curve segment displayed at two different magnifications, the selected critpts will not always be the same, and the selection at the lower resolution will not always be a subset of those at the higher resolution (e.g., Fig. 4). This is in contrast to the commonly held assumption that critpt assignment should be independent of scale of resolution.
4) A *threshold of significance*—a minimal level of criticality below which variations are considered to be noise and no critpt designations are made. (Some investigators reject the idea that any user supplied parameters or thresholds should be necessary.)

We have addressed the above issues through the solutions to the following set of subproblems.

1) Definition of an algorithmic procedure (which is parameterized to deal with noise and scale) for assigning criticality values to each point on a curve independent of decisions made about the locations of (other) critpts. The solution to this problem, essentially the procedure given

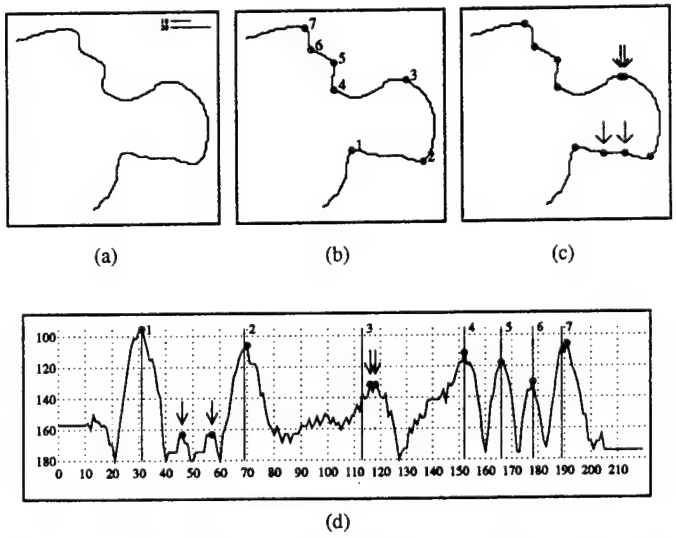
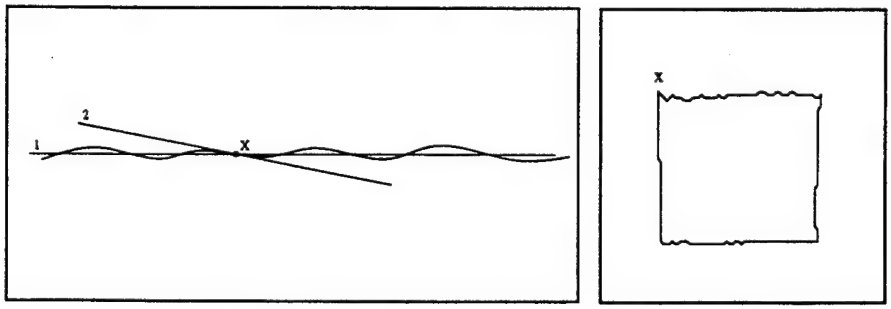


Fig. 3. Comparison of SSS and R/J-curvature evaluation on test curve 166. (a) Test curve 166. (b) SSS selected critpts. (c) R/J-curvature selected critpts. (d) Plot of R/J-curvature along test curve. Abscissa = sequence number of point on curve. Ordinate = angle (in degrees) computed at point. (Angle-arms are 10 units each for R/J-C; stick length is 20 units for F/B-S.

The continuous curve in (d) represents R/J-curvature along the test curve shown in (a). The vertical lines in (d) mark the sequentially numbered critpts selected by SSS as shown in (b). The critpts corresponding to the extreme values of R/J-curvature shown in (c) are marked as circles in (d). The arrows in (c), and in the corresponding locations in (d), illustrate anomalous selections using R/J-curvature.



(a) (b)

Fig. 4. Curvature and saliency are functions of curve resolution. As illustrated in (a) above, we can draw more than one visually acceptable tangent to many of the points on this curve at the given resolution. As resolution increases, tangent 2 would dominate at point x; as resolution decreases, tangent 1 would dominate at the same point. In (b), the angle at x can be seen as 45° at one scale and 90° at a larger scale. Thus, curvature and saliency are not unique properties of curve points.

in Fischler and Bolles [6], provides answers at a human level of performance for isolated critpts (i.e., along a section of a random curve, generated as described in Appendix A, for which human subjects select only one critpt). Thus, for the domains we experimented with (and especially the domain defined in Appendix A), we were able to assign fixed values to scale/resolution and noise/significance parameters so that our program would make the same selections as human subjects when there was near-unanimous agreement among these subjects. This algorithm is described in Appendix B.

2) An analysis of how geometric scaling of the input curve, and resolution-specific operations on the curve, can be equated, and thus the development of a basis for normalizing criticality scores across scale.
3) Development of a general approach to the problem of resolving the competition/cooperation interactions of geometrically related objects based on local dominance. The same machinery used to deal with interactions at a given scale of resolution is also used to resolve conflicts across different scales of resolution.

In the remainder of this paper, we describe our solutions to the problems enumerated above and then present examples and experimental results to justify the design decisions we made and to illustrate the performance capabilities of our algorithm.

IV. Evaluation of Saliency

Saliency is a critical attribute (for description and recognition) assigned to perceived things in the world by the human visual system (HVS). Although this is an elusive concept in general, task specific specializations of this concept are easily found that elicit consistent choices across human subjects. An acceptable computational definition of contour/curve saliency must provide the following.[6]

- The specification of a procedure that quantifies the abruptness and extent of the deviation of a curve from its straight-line continuation; a sharp bend is more salient than a shallow one, and the greater the excursion, the more prominent/salient the feature.
- Agreement with human judgement in terms of selection and accuracy of placement of the critical points (in some well-defined context).

A. Computational Definition of Saliency

Conventional definitions of curvature present a number of serious problems with respect to their use as a saliency measure in computational vision (CV). First, the mathematical definition is based on the properties of a curve in the infinitesimal neighborhood about the point at which curvature is being measured. For the finite precision quantized curves dealt with in CV, it has been difficult to find a suitable approximation to the limiting process originally intended for use on *mathematically continuous* curves. Second, it is readily observed that saliency is not an infinitesimal point property but is based on some finite extent of the curve. A proposed solution to both problems, offered by Rosenfeld and Johnston (R/J) [19] was to find an appropriately sized segment of the curve about the point in question and take a "snapshot" of the limiting process at this single (implied) scale. That is, rather than the rate of change of tangent angle with respect to curve length, R/J proposed measuring the angle between two fixed-length chords, where the lengths correspond to the computed "natural scale" of the curve about the given point. We will call this curvature analog the R/J-curvature. There are a number of other definitions of mathematical curvature (e.g., the limiting radius of a circle whose three defining points converge at the curve-point in question) which have analogs that could have been used in place of the angle measure in R/J-curvature, but as we show in Appendix C, these definitions are monotonically related and do not really present distinct alternatives. Thus, R/J-curvature is a suitable representative for the whole class of mathematical curvature-measure analogs.

In [6], our concern was not to find a good digital analog for curvature but rather to find an effective measure of saliency. The quantity defined in that paper can be viewed as a curvature-extremum measure in which the limiting process (in scale) is replaced by a scanning process (in space) more appropriate to digital curves. The scanning process is parame-

[6] In this paper we are primarily concerned with saliency based on *local* cues; locations on a curve where there is a transition from one type of curvature behavior to another, e.g., from perfectly straight to wiggly, may also be psychologically salient, but such forms of *global saliency* are beyond the scope of our current investigation.

terized by scale, and the resulting measure is a signed quantity which we call F/B-saliency (F/B-S).

Although the particular choice of a curvature measure as a component in a complete system for selecting the most salient points (critpts) on a planar curve depends on many factors, it is still interesting to compare the raw scores returned by *curvature analogs represented by the R/J-curvature* with the extreme points (ultimately) selected by our algorithm (SSS) as shown in Figs. 2 and 3 for a randomly generated curve. In these figures we observe problem situations that highlight some of the differences between the two underlying metrics (R/J-curvature and F/B-saliency).[7]

There are some problems with *any* raw measure of curvature that must be dealt with using procedures that invoke (at least) local *context*. For example, in Fig. 3 we see a case (double arrow) where two critpts were selected at almost adjacent locations along the curve. This undesirable behavior was not eliminated by the simple nonmaximum suppression filter that produced good results in most other situations. It is necessary to use more specific criteria in deciding when two critpts are too close together, and also what to do when the adjacent points have equal saliency scores (e.g., arbitrarily eliminate one of them or eliminate both and place a new critpt between them). In Fig. 3 we see cases (two single arrows) where almost invisible features were chosen as critpts because they *did* have locally extreme curvature scores. How do we decide when to reject such occurances? In Fig. 2 we see a case where a critpt (designated by an arrow) was inserted at a location displaced from the position we consider correct; this was due, in part, to the length of the arms of the angle measuring "operator" relative to the size of the feature (see Fig. 2(d))—it is not always possible (or practical) to find an appropriate operator size for every potential feature. In the following sections (and appendixes) of this paper we describe and justify the methods we employ to deal with these problems. The issue we are primarily concerned with in this section is the choice of a basic saliency metric. We justify our preference for the F/B-S metric on two grounds:

1) Unlike the fixed-scale mathematical (FSM) curvature analogs (e.g., R/J-curvature), F/B-S rarely makes an error in positioning a critpt, or in ignoring a salient point that human observers would select. The issue here is robustness. F/B-S integrates information over an extended set of "looks" at the curve segment containing the point whose saliency is being measured. FSM techniques take a single look at the situation. Thus, our main problem with the F/B-S metric is selecting the most salient of the selected critpts to be retained as our final result (the filtering operation generally involves the elimination of less than half of the points originally selected).
2) The F/B-S metric is responsive to both the curvature and the size of a curve feature. This provides a common basis for ranking critpts at a given scale (so that the larger of two geometrically similar objects is assigned a higher saliency score) as well as across scales by taking into account the size of the operator. The FSM-curvature analogs are insensitive to the size of the feature—they inherit the mathematical property that curvature is a point property and only the smallest neighborhood about a point that allows us to measure curvature is relevant (this implies a single natural scale at any point on a curve, a concept we reject, e.g., see Fig. 4).

[7] In both of the figures, we used fixed common scale parameters for both metrics as noted in the figure captions. It should be remembered that R/J-curvature, as we define it in this paper, is representative of a whole class of curvature-based metrics and is not intended to duplicate the complete Rosenfeld/Johnston algorithm—they also incorporate a procedure for finding a preferred stick length. However, many of the problems with the performance of the complete algorithm, which are discussed in [4] and in other of the papers we reference, can be observed in the performance of the R/J-curvature metric.

B. Comparison of the Saliency Selection System (SSS) with Human Performance

The primary criterion for judging the competence of the overall saliency selection system (SSS) we present in this paper is its ability to match human performance—both in the defined task and with respect to generic evaluation of the selected critpts. We performed a set of informal experiments with 11 human subjects (also see the experiments described in [6]). The instructions given to the subjects and the resulting selections are shown in Fig. 1. We also show the selections made by the SSS algorithm. The results of these (and additional but not described) experiments can be summarized as follows

- At least 9 of the 11 subjects selected the same set of six or more critpts on each of the four curves we used in the experiments, and the SSS chose the same set of critpts. Every critpt selected by the SSS was also selected by at least one human subject.
- In spite of the high degree of consistency in the overall selection of salient points, the human subjects differed in the order in which they chose these points. We tried a number of experiments in which the only difference was a very slight change in the wording of the instructions and obtained different orderings (across the same set of selected points) from our subjects. It is obvious that the subjects used a global strategy to match the task (different for each subject) in choosing the order in which the points were selected—even though the specific points selected were largely determined by local context.

In addition to the curves used in the human experiments, we ran the SSS algorithm on (the order of) 1000 randomly generated curves with no obvious errors. Fig. 5 shows the results of a (typical) sequence of 40 consecutive experiments.

V. Dealing with the Problems of Scale and Resolution

A vision system, concerned with creating a description of some object that may be encountered again in the future, perhaps when the object is closer or farther away, must take scale or magnification into account when deciding what shape elements to pay attention to. Under extreme changes in resolution, when salient features might appear or disappear, it may not be possible to make an informed judgement in the assignment of relative saliency scores; but for a limited range about a given resolution, this should indeed be possible.

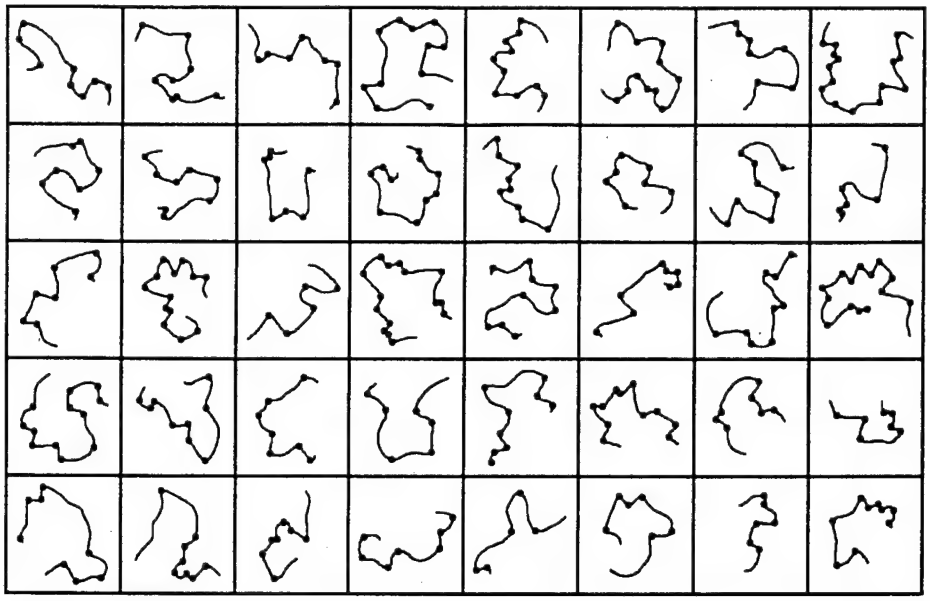


Fig. 5. Critical points found by the SSS algorithm for a set of 40 random curves.

Obviously, geometric properties of objects that are invariant over scale are especially valuable in describing and recognizing the objects, since absolute scale is often impossible to judge in an image, and even relative scale can be difficult to describe or measure if the measurement must be referenced to the global geometry of the object. One of the main issues we address in this paper is how to define extrema in the bending of a curve as a local effectively scale-invariant property that is in agreement with the judgement of the human visual system.

If we define criticality of points on a digitally represented curve in terms of quantities that have dimensions that must be measured by some physical process, then there is no direct way of invoking such formally defined mathematical concepts as the derivative, or curvature, which require limiting processes of infinite resolution. Approximations to these concepts are resolution dependent (e.g., the size of the operator employed), and measurements made on most objects will not scale in any simple or uniform way. Further, if we examine a curve through a fixed-size window (either a fixed region of a computer screen or the foveal region of the human retina) and successively increase the resolution at which the curve is displayed, some of its parts will eventually disappear from view and some of the smaller original structures, which were not significant, will now dominate the visible appearance of the curve (e.g., Fig. 4).

If the mathematical definition of curvature were applicable to digital imagery, then many (but not all) of the issues of scale could be resolved. There is still the problem that a very small glitch can have a very high value of curvature but a very low psychological significance. Thus the scale or size of a feature (e.g., the glitch) is an issue. The term "feature" does not appear in our problem definition; in fact, by focusing on local curve properties, we had hoped to eliminate the need to invoke this concept since an appropriate definition is far from obvious.[8] Since scale can't be ignored (even if we had a good approximation for curvature in the digital domain that was independent of scale) the following questions arise:

- The distinction, if any, between resolution and scale;
- How to choose a range of scales appropriate to the specified performance criteria;
- How to measure criticality at different scales;
- How to compare criticality values computed at different scales;
- The relationship between smoothing and scale change;
- The relation between operator size and scale change;
- How to make cooperation/competition judgements across scales; and
- How to determine the features for which we expect consistency (of criticality scores) to hold across scales, and where such consistency can't be expected (if the latter were never the case, we could always do our analysis at one scale and compute the criticality values at other scales as needed).

Although consistency at all scales and for all features is not possible, over some range of scales (say 5:1) we expect there to be a normalization factor that allows us to compare the saliency scores computed at one scale with values computed at other scales. We would also expect that relative locations of local extrema for *certain* features would remain fixed as a curve is scaled, regardless of the size/scale of the operator that assigns the criticality scores.

Some of the earliest work (e.g., Rosenfeld and Johnston) on finding salient points merged the problem of assigning a curvature measure to a point with that of determining the

[8] Intuitively, there are sections of any given curve that we call features; these entities provide the psychological basis for the selection and relative saliency of the associated critpts. Critpts are markers that define the shape and boundary of features; the extent of the curve corresponding to a feature will generally subsume the region of support for the curvepoints comprising the feature. Features can overlap, and their boundaries are not always apparent.

scale at which to measure curvature. The key idea is that each point has a single scale at which its curvature should be measured; this scale is usually found by a search process over successively larger scales until some measured quantity achieves a local extremum.

A. Change of Scale versus Change of Resolution

If we magnify a continuous curve that was originally represented at infinite precision, every point of the new image corresponds to a point in the original image, but its x- and y-coordinate values have been multiplied by some real number, which we will call the scale factor. No information was introduced nor lost, but the physical space required to render the curve has increased. However, if the original curve was represented at finite resolution (e.g., each point as a pair of integer coordinates), then (say) doubling the scale leaves us with a disconnected set of points. Filling in the gaps requires introducing new information. Here we will say that a change of resolution has occurred (a change in resolution can also result in the loss of information, as in the case of demagnification or smoothing at some fixed resolution). Thus the concept of a scale change corresponds to a reversible transformation while, in general, a change in resolution involves an irreversible process in which information is lost (as in smoothing) or new information is introduced (as can occur in zooming).

If we compute the curvature for points on a continuous (infinite resolution) curve at two different scales, we will generally get two distinct sets of values (e.g., a circle with radius 2 is a scaled version of a circle with radius 1, but by definition their curvatures are in the ratio 1:2. On the other hand, the angles of a triangle remain unaltered under a scale change). It will be the case, however, that for smooth curves the local extrema will be found at corresponding locations—but even here, the numerical values of curvature will not scale in any simple way (curvature is a nonlinear function).

B. SSS Mechanisms for Evaluating Saliency at Different Scales and Resolutions

In designing a computational module to evaluate saliency subject to the ideas discussed above, we can pursue at least three distinct strategies:

1) Assume that saliency is independent of scale, or that there is a natural scale associated with each location on the curve that must be discovered.
2) Use a fixed scale saliency measure, but generate multiple versions of the given curve at some predetermined set of scales.
3) Parameterize the saliency measure to give results approximating those that would be obtained from strategy 2 for the selected scales.

We previously argued against strategy 1 on the assumption that a unique natural scale *cannot* generally be associated with a single curvepoint (see Fig. 4). We have chosen strategy 3 since strategies 2 and 3 are conceptually compatible, but 3 could be computationally more efficient if we can find a simple way to use some combination of operator scaling and score normalization so that both approaches give (nominally) the same scores in most situations. Intuitively, doubling the stick length (in the F/B-S metric) for a simple convex section of a curve should result in four times the score assigned to the corresponding critpt: The stick is now positioned twice the distance from the critpt in most of its "looks" (i.e., placements of the stick which subsume a curve segment containing the critpt), and there are twice as many looks. Thus, the procedure we employ, *normalizing all scores by dividing by the square of the stick length,* will leave invariant the saliency scores assigned to features that should be scale invariant, such as the angle formed by two (effectively) infinite straight lines; Appendix D provides a more complete discussion of this point. On the other hand, for those features that have limited extent along the curve, comparable to the scales we wish to discriminate among, the larger scaled versions of the features will be assigned higher scores. Fig. 6 shows the correspondence between scale parameterization of the F/B-saliency measure versus scaling the curve itself for some example curves.

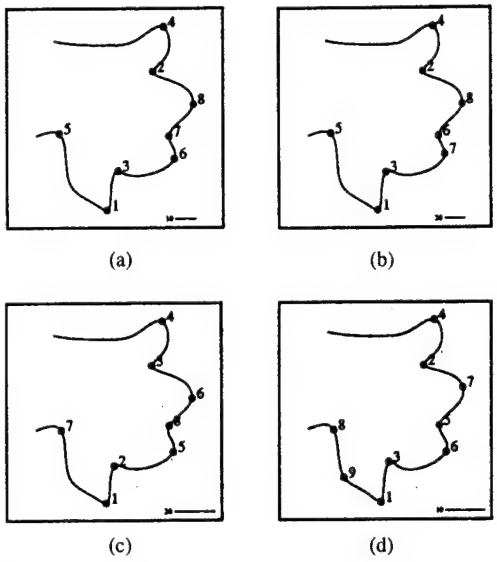


Fig. 6. Comparison of curve resolution versus change in operator size (stick length) on test curve 37f. Critpts are labeled with rank ordering of their scores. (The scaled curves are drawn to fit into a standard-size box.) (a) Scale 1; stick length 10. (b) Scale 2; stick length 20. (c) Scale 1; stick length 20. (d) Scale 0.5; stick length 10.

VI. Cooperation/Competition Interactions Between Critical Points

An important contribution of this paper over the work presented in [6] is a major revision of the approach to filtering the critpts, based on comparisons at a given scale as well as across different scales. At a conceptual level, there are two main differences.

First, in the earlier work we did not use the information about the sign (concavity/convexity) of the computed F/B-saliency; in our current algorithm, we separate all the candidate critpts into two sets corresponding to positive and negative

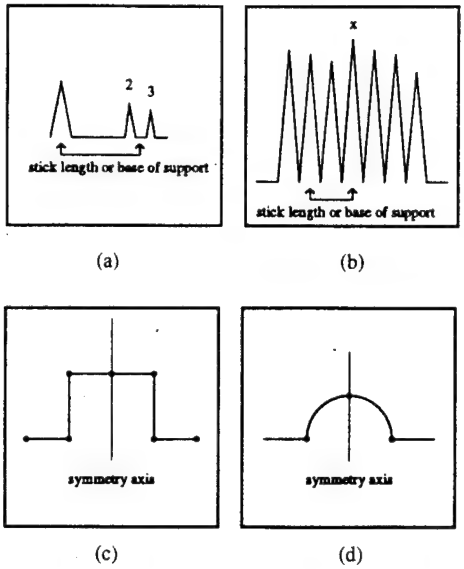


Fig. 7. Problem situations correctly handled by the SSS algorithm. The critpts at peaks 2 and 3 should be both accepted or both rejected in (a). In (b), the critpts at all peaks surrounding the peak at x are approximately equal in saliency; they should all be accepted or all rejected. The pattern of accepted critpts should be symmetric in (c) and (d).

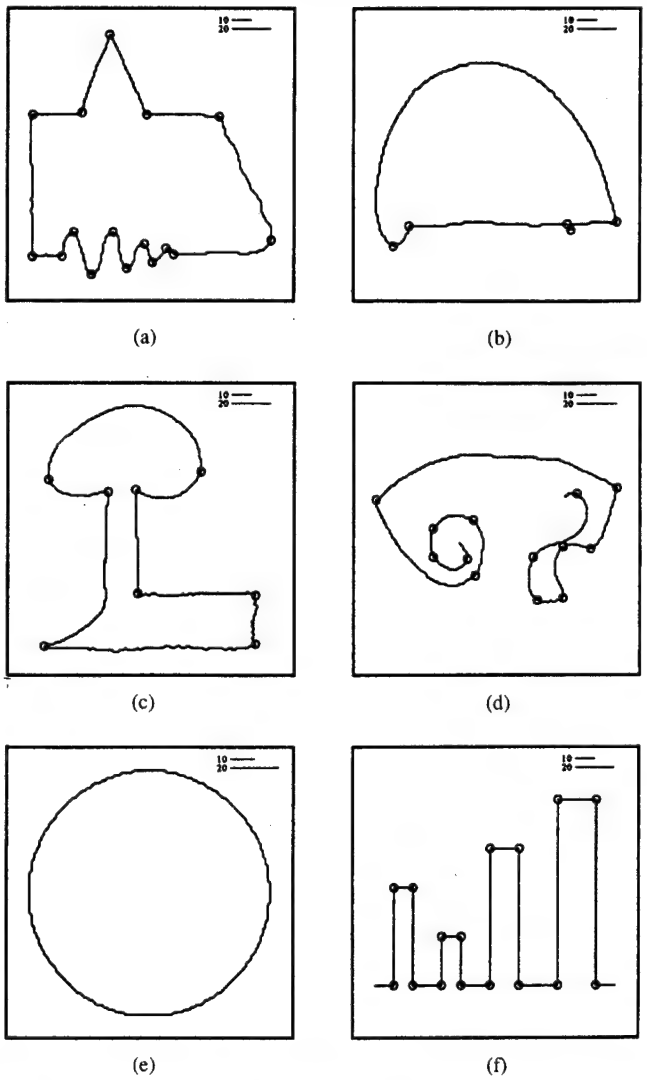


Fig. 8. SSS selected critpts on various curves with global structure. (a) 455 curve points, 16 critpts. (b) 335 curve points, 5 critpts. (c) 503 curve points, 8 critpts. (d) 425 curve points, 13 critpts. (e) 282 curve points, 0 critpts. (f) 606 curve points, 16 critpts.

F/B-S.[9] These two sets are processed independently of each other (by identical procedures), and the resulting selections are combined by logical union to produce the final output. Our own observations confirm those of other researchers (e.g., Hoffman and Richards), that positive and negative curvature extrema appear to be distinguished from each other by the HVS, in part because they play different roles in partitioning and description tasks.

Second, in the earlier work we used a simple dominance criterion for competition of closely spaced critpts detected at different scales.. A critpt detected at some given scale would suppress all critpts detected at smaller scales (shorter stick length) that were located within a specified scale related distance from it. This rule rarely produced "ugly" errors, but occasionally it caused the obviously correct critpt to be deleted in favor of one slightly displaced from the preferred location. A significant portion of the work described in this paper has been focused on finding a more effective and uniform basis for establishing local dominance. In other sections of this paper we provided a justification for a normalization factor that would permit us to assign a saliency ranking to competing critpts, regardless of the scale at which they were originally detected. Thus, competition, both within and across different scales is now treated in a uniform manner. In the following subsection we discuss some of the specific problems that must be resolved in competition resolution, and the algorithmic procedures we invoke to deal with these problems.

A. Mechanisms for Filtering Competing Critpts

As already noted (previous subsection), there is no competition between points having differently signed saliency scores and scores computed at different scales (i.e., with different length sticks) are normalized to make them directly comparable. In Fig. 7 we present a number of difficult problem situations we must correctly resolve.

In Fig. 7(a) and (b) we see situations where nonmaximum suppression[10] over any fixed interval will eliminate some critpts but will accept adjacent critpts of approximately equal saliency to those eliminated—this form of filtering is not consistent with selections made by human observers and would be unacceptable behavior in an algorithm intended to perform at a human level of competence.

In Fig. 7(c) and (d) we see some additional situations, which are not likely to occur in our experimental domain (i.e., curves generated by the procedure described in Appendix A), but which we would nevertheless like to handle correctly. In both of these examples, we expect to get a symmetric arrangement

[9] For an open curve segment, the assignment of positive versus negative is arbitrary; the important consideration is that we use the information about the direction of deviation of the curve from the stick to separate detected critpts into the two possible categories, which are then processed separately.

[10] Nonmaximum suppression refers to the procedure in which each critpt suppresses other potential critpts, with lower saliency/criticality scores, falling in its region of support.

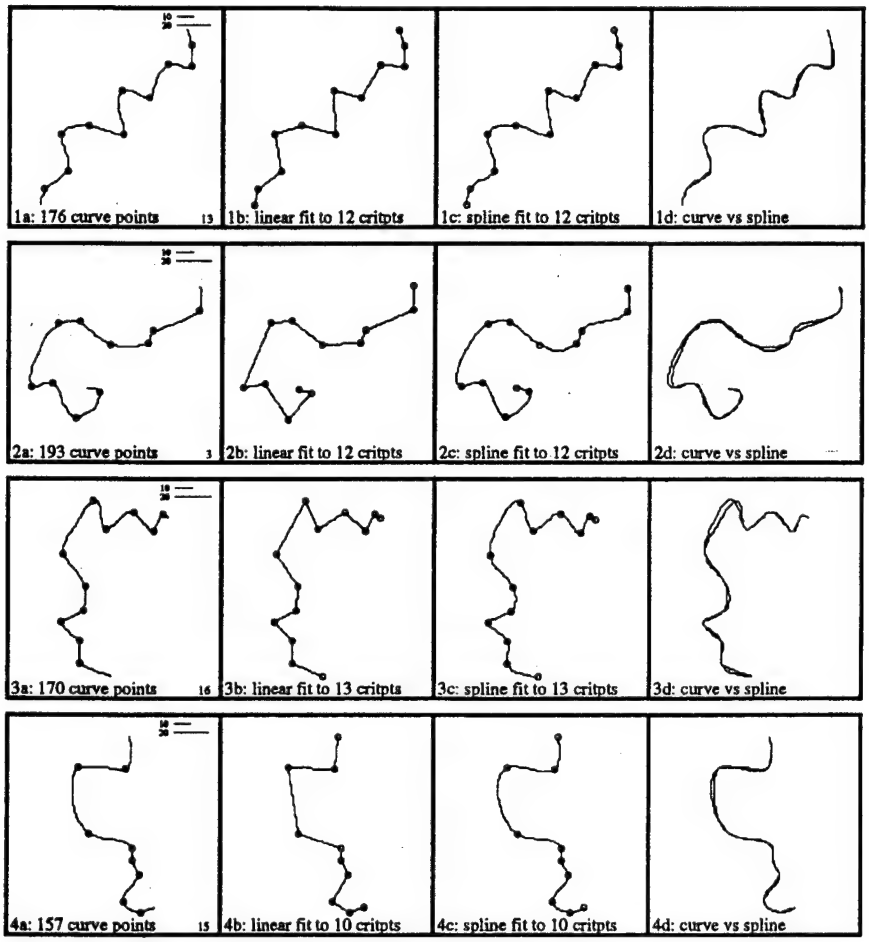


Fig. 9. Examples of curve reconstruction based on using straight line and spline interpolations of critpts. (a) Input curve showing SSS selected critpts. (b) Straight line interpolation of critpts. (c) Spline interpolation of critpts. (d) Comparison of original and reconstructed curves.

of accepted critpts; we require correct filtering of critpts with (approximately) equal saliency scores, even though they are separated by significant curve-length intervals.

One of the algorithmic mechanisms we devised to deal with the above problems (described in greater detail in Appendix B) is to construct an array with one slot for each indexed location along the curve (actually two such arrays, one each respectively for positive and negative saliency scores). Each slot is either free or owned by exactly one critpt. A critpt occupies only one of the slots it owns; this occupied slot corresponds to its actual location along the curve. A "new" critpt[11], contending for a slot, must have a normalized score greater than the existing value stored in the slot to capture it. If a new critpt captures a slot occupied by (as opposed to simply being *owned* by) a previously dominant critpt, all of the slots of the now-dominated critpt are also captured. This mechanism provides a way of avoiding the need to choose a fixed-sized base of support for a critpt and successfully deals with the problem situations depicted in Fig. 7(a) and (b). The additional algorithmic mechanisms introduced to correctly handle the situations of Fig. 7(c) and (d) are discussed in Appendix B.

[11] All the potential critpts are detected, sorted, and then entered into the array in increasing order of saliency to avoid sequence dependent effects.

VII. Algorithm Performance

The algorithm discussed in the previous sections of this paper, and described in detail in Appendix B, has been compared with human performance (Fig. 1) and has been run on hundreds of randomly generated images (as described in Appendix A) without making any obvious errors. In all these cases the same set of parameters was used with no operator involvement. Fig. 5 shows 40 consecutively generated random curves and the critpts selected by the algorithm. Fig. 8 shows additional examples (also using the standard parameters) that illustrate performance results on curves with characteristics outside the specific experimental domain that was our main focus. Fig. 17 in Appendix E shows results of the algorithm run on curves extracted from real images.

In Attneave's original paper, and in a number of recent publications (e.g., Imai and Iri; Teh and Chin) critpts were used as the basis for reconstructing a curve using straight line and spline interpolation. That is, the critpts were a highly compressed representation of the qualitative information stored

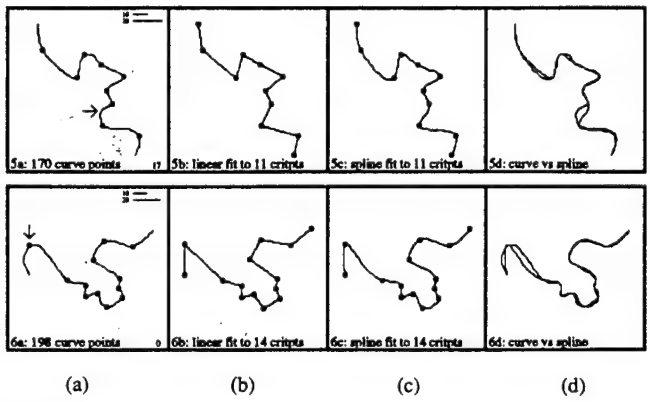


Fig. 10. Examples or curve reconstruction based on using straight line and spline interpolations of critpts. (a) Input curve showing SSS selected critpts. (b) Straight line interpolation of critpts. (c) Spline interpolation of critpts. (d) Comparison of original and reconstructed curves.

in the original curve that allowed a recognizable version of the curve to be reconstructed by a simple algorithmic procedure.

In Figs. 9 and 10 we show some typical examples of curves regenerated from the critpts extracted by the SSS algorithm. In these cases the noise threshold was reduced to the lowest level permitted by our current implementation, but all other parameters were left unaltered. The spline reconstructions were found to be very good in general, even though the SSS algorithm was not designed to be used for this application. The two types of error that occur are illustrated by curves 10(5) and 10(6). In curve 10(5a) an arrow shows where an additional critpt could have been added to eliminate the discrepancy that is apparent in 10(5d); the SSS algorithm found a potential critpt at the location of the arrow but subsequently eliminated it because its saliency score fell below the preset (standard) significance level. In Fig. 10(b), the single critpt at the location of the arrow doesn't provide enough information to reproduce the curve as precisely as desired at this location.

It is obvious that we could have designed the SSS algorithm to produce as good a reconstruction as desired by performing the comparison operation depicted in Figs. 9 and 10, and accepting enough additional critpts to eliminate any significant errors. In this case we would get better reconstructions at the cost of choosing points that would *not* be selected by most human observers. It is interesting to note that even though the nominal task is curve reconstruction, there appears to be a human criterion for saliency/criticality that falls a bit short of what seems to be required for curve reconstruction without noticeable error.

VIII. Open Issues and Unresolved Problems

The problem posed at the beginning of this paper (*devising a procedure to find salient points on reasonably well behaved digital curves that have no global structures*) appears to be solved by the SSS algorithm—at least for curves that are geometrically similar to those formed by the process described in Appendix A. Where does this leave us in general? Obviously, the issue of recognizing global structures is still open, but the SSS algorithm still does very well on most examples of unconstrained curves (e.g., see Fig. 8).

If we look more closely at the problem of *local* saliency detection, there are a number of issues that are still not completely resolved—most having to do with the digital grid representation for curves and the choice of a distance metric for digital curves. The most important of these open issues is the anomalous behavior of the F/B-S metric in the presence of very sharp angles. As noted in Appendix D, F/B-S peaks for angles between 45° and 60° and then falls for sharper angles. To the extent that this is a matter of concern, the problem appears to be easily fixed by the *augmented* F/B-S metric included in Appendix B; however, the augmentation for sharp angles has not been fully tested.[12]

Some additional issues we did not fully explore are: 1) stability of critpt placement when curves are rotated on the digital grid, 2) the use of floating point versus integer representation for curvepoint coordinates, and (3) the relative merits of Euclidean versus curvelength distance for stick-length determination. All of these matters were at least casually examined, and none appeared to pose a significant problem or present an opportunity for significantly improved performance.

There is also a set of open issues associated with the proper selection of a set of stick lengths for saliency determination, and the stability of critpt placement and ranking when small changes are made in the selected stick lengths. Earlier in this paper we argued that stick length is a problem-dependent parameter. Critpts selected by a stick whose size is much smaller or larger than the associated feature will have reduced scores; thus stick length provides a way of specifying the size of features that are of potential interest.

With respect to stability, we would expect that over a continuous range of stick lengths the critpts would remain relatively fixed in location and ranking, with jumps occasionally occurring in both of these attributes—and indeed, we found that this is the case for most of the curves we examined (e.g., see Fig. 11 as a typical example). The one exception occurs when a long segment of a curve has slowly changing curvature, and here it is not clear where the critpts should be positioned—even for human observers. For example, in Fig. 1, column 4, human subjects were inconsistent in the number and placement of their selected critpts on the lower left lobe of the curve. In Fig. 12 we show the performance of the SSS algorithm in choosing critpts on a smooth curve with slowly changing curvature, and even though there is some undesirable drifting in critpt placement there is no obvious error in any of the selected placements.

Stability of the selected critpts with respect to minor perturbations in the shape of the curve is another important issue

[12] The problem of very sharp angles did not arise in our primary experimental domain for two reasons. First, on the digital grid, it is impossible to construct an ideal angle smaller than 45° without aliasing. But the second and more direct reason is that the minimum spanning tree (MST) technique we employed, for constructing the random curves, will not produce very sharp angles. Theoretically, the MST will not produce an angle smaller than 60°, but because of digital effects and other processing steps angles as small as 45° can occur. The authors of this paper have developed procedures for extracting linear structures from imagery using a technique that employs the MST as an initial filter (see [7]). For curves obtained by such a procedure, sharp angles are eliminated.

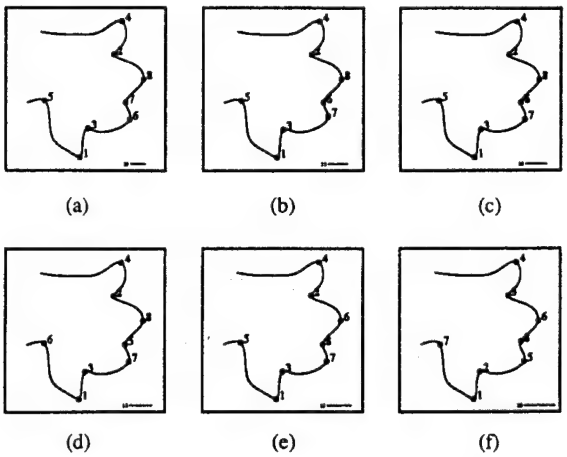


Fig. 11. Behavior of F/B-curvature with changing operator size (stick length) on test curve 37f. Critpts are labeled with rank ordering of their scores. (a) Stick length 10. (b) Stick length 13. (c) Stick length 14. (d) Stick length 15. (e) Stick length 16. (f) Stick length 20.

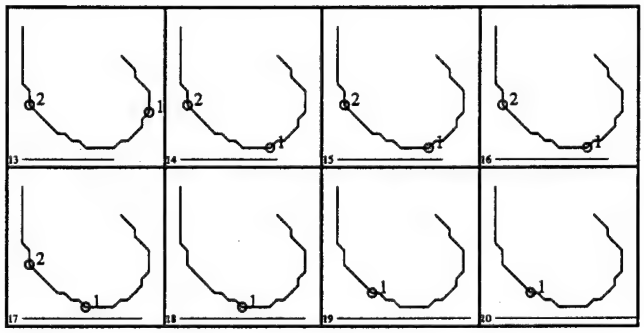


Fig. 12. Behavior of of F/B-curvature with changing operator size (stick length). Critpts are labeled with rank ordering of their scores.

but is difficult to quantify in a general setting; one must first provide a criterion for deciding when a perturbation is minor. Although we did not attempt to formally explore this issue, a few casual experiments indicated that the algorithm is robust in this regard.

IX. Discussion

Curve partitioning is an active research area that not only is of theoretical interest as a basic element in pictorial description (see, e.g., Attneave; Bengtsson and Eklundh; Hoffman and Richards), and for providing insight into the partitioning problem in general (see, e.g., Fischler and Bolles), but has many potential applications. Some of the more immediate ones include data compression by using critpts as the basis for regenerating a curve by straight line or spline interpolation (see, e.g., Imai and Iri; Teh and Chin), matching/recognition using critpts and/or the partitioned curve segments (see, e.g., Mokhtarian and Mackworth; Wuescher and Boyer), and as a key component of an interface for man–machine communication about pictorial objects (the ability to point at icons representing symbolic objects has revolutionized the computer-user interface; to extend this capability, one would like to be able to point to a location in an image and have the machine be able to deduce the component being referred to—image partitioning in general, and especially curve partitioning, are critical to this goal).

In this paper we have focused on one specific aspect of the curve partitioning problem: duplicating human performance in the selection of a small number of points (called *critpts*) along a curve segment that could be used as the basis for *reconstructing* the curve at some future time. Although there will generally be a significant degree of overlap in the points selected by the techniques referenced above (focused on different applications), there are also significant differences. There has been very little recent work on the generic problem of choosing psychologically salient points with which to directly compare our results. On the other hand, we have conducted a relatively large number of experiments with uniformly good results (e.g., see Fig. 5).

There are two major paradigms[13] underlying the published work on partitioning planar curves. The first involves obtaining a mathematically differentiable representation of the given digital curve using splining or Gaussian convolution (see, e.g., [14]). This gives good results for many applications, but the salient points on the *smoothed* curve are often displaced from their original locations (or eliminated). This paradigm is not suitable for our purposes in this paper.

The second paradigm, which includes the work described here, is to first measure some approximation to the curvature at each point on a curve. This usually involves choosing, or finding, an appropriate scale at which to make the curvature measurement. This is typically accomplished by making the curvature measurement over increasingly larger curve segments (centered on the curve point being evaluated) until either the computed curvature at the point or some related quantity reaches a local extrema. Each point is assigned a saliency/criticality value (its estimated curvature) and an interval length along the curve centered on the point (called its *region of support*). The region of support is then used for nonmaximum suppression—each point suppresses other points with lower criticality scores falling in its region of support.

Major differences between our approach and other work under this second paradigm include:

- A generic saliency measure that often selects points corresponding to local curvature extrema, but which in many situations is in better accord with human selection preference and placement accuracy.
- A distinct approach to the problem of dealing with curve features salient at different scales. The conventional approach is to associate a single scale with each curve point, which in turn defines a fixed region of support to be used for nonmaximum suppression. In our approach, we measure the saliency of each curve point at a number of different scales and have developed procedures for allowing potential critpts, found at different scales and spatial locations to compete[14] with each other. This competition

[13] Additional approaches are available for partitioning 1-D curves; for example, see [7] or [22]. As noted in Appendix B, the 1-D partitioning technique in [7] is used as a component of the SSS algorithm.

[14] It is interesting to note that we have not found a use for cooperative reinforcement; cooperation appears to be a global relation. Competition is important at the local level (e.g., lateral inhibition).

is not restricted to any fixed extent of the curve (which thus avoids anomalous selections caused by an important event occurring just beyond the fixed limit of search, i.e., the *horizon* effect).

Our approach to local saliency selection can be considered a form of automated preattentive perception. Potential extensions could include dealing with more global curve features, such as recognizing the intersection of extended straight line segments, or transition points between analytic curves with different parameters, or global symmetries and repeated structure. Recognizing these more global structures, and ranking them with respect to human perceived saliency, may well fall outside the competence of the basic approach described in this paper.

Appendix A
Generation of Random Curves

The following method was used to construct the random curves used in the experiments described in the body of this paper.

1) Thirty (x, y) pairs are generated for each curve. Each value of x and y are generated by a uniform-distribution (0–1) random-number generator and then multiplied by 100 to produce numbers (coordinate values) uniformly distributed between 0 and 100.

2) The 30 points are next linked by a minimal spanning tree (MST).

3) A diameter path is extracted from the MST, and the ordered subset of the original randomly generated points that fall along this diameter path are the input sequence provided to a spline-fitting routine [3], which returns a continuous curve represented by a sequence of (x, y) coordinate pairs. These sequences, typically containing on the order of 150–250 points, are the random curves used in our experiments.

Appendix B
An Algorithm for Computing Curve-Point Criticality

The partitioning algorithm described in [6] has been modified as discussed below.

The algorithm collects candidates (peaks) for the critical points of a curve by examining the deviation of the points of the curve from a chord or stick that is iteratively advanced along the curve. Sticks of different lengths are used to find critical points that are salient at different natural scales on the given curve. (Except when explicitly stated otherwise, two sticks were used for all the experiments discussed in this paper; one of length 10 pixels and the other of length 20 pixels.) The algorithm provides the option of using arc length along the curve, or the Euclidean length of the stick, to determine the separation of the endpoints of the stick on the curve; we used the Euclidean length of the stick for all of the experiments discussed in this paper. One end of the stick is advanced along the curve, one pixel at a time, and the other end is placed at the first (sequential) position further along the curve for which the Euclidean distance equals or exceeds the specified stick length.

For each placement of the stick, an accumulator associated with the curve point (in the interval of the curve between the two endpoints of the stick) of maximum deviation from the stick is incremented by the absolute value of the distance from the point to the stick if this distance exceeds a predefined noise threshold. However, for the given stick placement, if there is more than one excursion (exit and return) outside the noise region, the underlying model is violated and the accumulators are not incremented. (The noise threshold was uniformly set to 20% of stick length; thus a Euclidean deviation of more than 2 pixels from a stick of length 10 was required to cause any modification of the associated accumulator.)

To deal with direction-dependent effects, a complete traverse is made in both directions along the curve summing the results in the same accumulators. The points that have locally maximum scores in the accumulators (called peaks) for any of a given set of sticks are the points from which the critical points will be selected.

The following information is collected for each peak and used to find the critical points.

- INDEX: The sequence number along the curve of the point at which the peak was located.
- STICK: The length of the stick (in pixels) used to find the peak.
- DEV: The sign of the deviation of the peak with respect to the curve.
- NSCORE: The normalized score, which is the score in the accumulator for the peak divided by the square of the stick length.
- TSCORE: NSCORE incremented by a small tolerance. $TSCORE = (1.01)NSCORE$.

The peaks are divided into two groups with like-signed deviation DEV. The critical points for the two groups are found independently of each other, and their union is returned as the set of critical points for the curve.

In finding the critical points, we stipulate that each peak's score has a region of support, *plus* and *minus* half its associated stick length, on each side of its position along the curve. An array (the support array) equal to the length of the curve is used to store the support information. The support information for a peak is a list (score INDEX STICK). For each peak, the support information may be entered at every index location covered by the region of support, depending on what was previously stored in the location.

When information about a new peak located at INDEX* on the curve is to be entered into the support array at some permissible location, the first task is to decide what score should be included in the information. Normally, it would be the list (NSCORE* INDEX* STICK*). But if there already is an entry at INDEX* and its score is > TSCORE*, the adjusted data is the list (TSCORE* INDEX* STICK*).

The element at INDEX* (where the new peak occurs) in the support array is treated specially when entering the information for a new peak. If there is an existing entry at INDEX* in the array, and its score (call it CSCORE) is > NSCORE* and ≤ TSCORE*, the peak information (CSCORE INDEX* STICK*) will be stored at INDEX* in the array.

For all other elements in the support region for the new peak in the support array, an element at J is replaced by the information for the new peak if there is no previous entry in the array or if the score in the adjusted data list described above for the new peak is greater than the score in the existing entry in the array. In addition, if the entry J is being replaced and J is also the INDEX for a peak that was entered previously, the support information for the new peak replaces the support information of the old peak wherever it occurs in the support array (i.e., even outside the new peak's original support region).

The following table shows the support information for a new peak if there is an existing entry in position J of the support region that needs to be replaced. (The reason for this additional complexity is to avoid the use of "hard" thresholds.)

Comparison at INDEX*	Entry for INDEX*	Entry for $J \neq$ INDEX*
CSCORE < NSCORE*	(NSCORE* INDEX* STICK*)	(NSCORE* INDEX* STICK*)
NSCORE* < CSCORE ≤ TSCORE*	(CSCORE INDEX* STICK*)	(NSCORE* INDEX* STICK*)
CSCORE > TSCORE*	No change	(TSCORE* INDEX* STICK*)

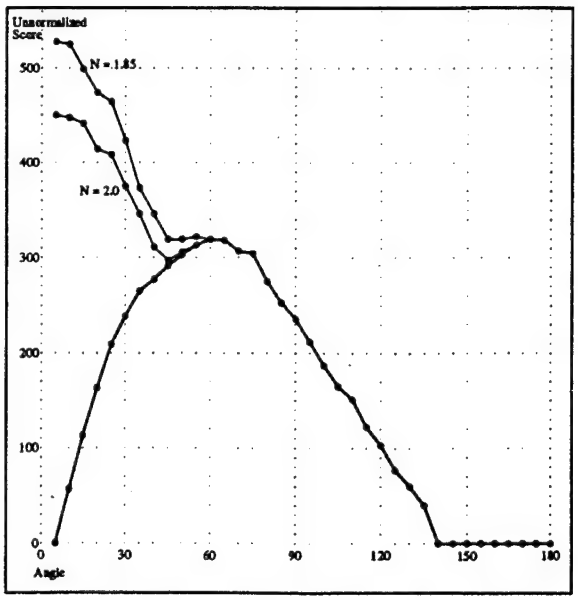


Fig. 13. Augmented and standard (unnormalized) SSS scores for angles at increments of 5° (stick length 20). The bold curve shows the standard SSS scores. The thin curves show the augmented scores for two different values of N at which we alter the way distance is measured from a curve point to the stick (i.e., when the number of curve points between the end points of the stick is greater than N times the stick length, the distance from the curve point to the midpoint of the stick is used as the deviation of the curve from the stick).

After the above processing, the critical points for the curve are designated as those points whose index into the support array equals the index stored in the information list of the array element.

It can be seen that the order in which peaks are entered into the support array can affect the final selection of the critical points because a peak's region of support can be altered by the capture process and thus depends on the state of the support array at the time the peak is entered. In our implementation of the algorithm for running the experiments, we entered the peaks into the support array as soon as they were computed in order to gain computational efficiency and simplicity, and we still obtained excellent results. From a purely theoretical viewpoint, a more principled and consistent procedure would have been to first collect all the peaks for all the sticks, then sort the peaks by their normalized scores, and finally enter them into the support array in order of increasing score.

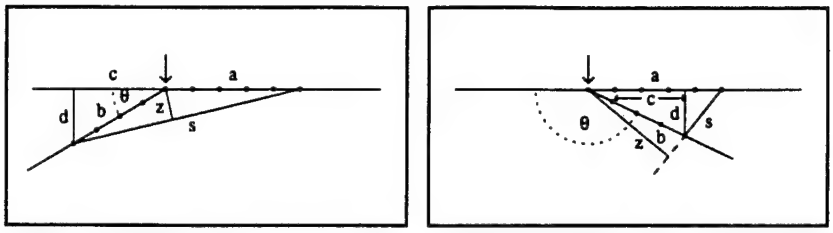
A completely separate postprocessing step, described below, is used to resolve the question of critical points that are close together in score and vicinity after the above operations.

Augmentation of the Basic Algorithm: There are some other details (of lesser theoretical interest) about the algorithm that were not explicitly mentioned in the text of this paper, or described above. For example, a special routine is used to eliminate all but the maximum element (or the central element in a run of equal values) in a dense sequence of (immediately adjacent) peaks. This step is performed after the peaks are detected but prior to entering them in the support array. This filtering procedure, identical to the 1-D ridgepoint algorithm described in [7], can properly handle the situations depicted in Figs. 7(c) and (d).

To deal with the problem of competing critpts of approximately equal saliency scores, we make three attempts to resolve the conflict by successively increasing (one unit at a time) the stick size of the smallest stick involved in originally choosing the competing points. If one of the larger sticks produces a single point in place of the original set of competing points, we accept the single point. If no acceptable solution was found after three tries, we accept all the original critpts (except when two critpts are so close together that they are not visibly separated, a distance of less than 5 pixels, here the program arbitrarily eliminates one of these two points).

To deal with the problem of very sharp angles (as discussed in Section VIII and in Appendix D) we detect situations in which the curvelength distance between the two endpoints of an advancing stick is at least twice the Euclidean distance between these same two endpoints; in these situations we measure the perpendicular distance from each subsumed curvepoint to the stick (as in the basic algorithm) and also measure the distance from each subsumed curvepoint to the center of the stick; the larger of the two distances is used to update the accumulator associated with the curvepoint. As shown in Fig. 13, this modified distance metric eliminates the anomalous behavior of the basic distance metric (see Table I) in which the score in the accumulator for the vertex point of an ideal angle (of less than 60°) monotonically drops to zero as the angle decrease to zero degrees.[15]

[15] As shown in Fig. 13, switching to the modified distance metric when $N = 1.85$ rather than when $N = 2$ would produce a smoother curve. However, it is not clear that there is any significant performance difference based on which of these two values is used; we ran some equally successful tests using both values.



(a) (b)

In Figure 14(a):

$$\begin{aligned} d^2+(a+c)^2 &= s^2 \\ (a+c)^2 &= s^2-d^2 \\ a+c &= \sqrt{s^2-d^2} \\ a &= \sqrt{s^2-d^2}-c \end{aligned} \tag{1}$$

$$\begin{aligned} \frac{s}{d} &= \frac{a}{z} \\ z &= \frac{ad}{s} \end{aligned} \tag{2}$$

Substituting (1) into (2):

$$z=\frac{ad}{s}=\left(\sqrt{s^2-d^2}-c\right)\frac{d}{s} \tag{3}$$

In Figure 14(b):

$$\begin{aligned} d^2+(a-c)^2 &= s^2 \\ (a-c)^2 &= s^2-d^2 \\ a-c &= \sqrt{s^2-d^2} \\ a &= \sqrt{s^2-d^2}+c \end{aligned} \tag{4}$$

$$\begin{aligned} \frac{s}{d} &= \frac{a}{z} \\ z &= \frac{ad}{s} \end{aligned} \tag{5}$$

Substituting (4) into (5):

$$z=\frac{ad}{s}=\left(\sqrt{s^2-d^2}+c\right)\frac{d}{s} \tag{6}$$

We note that c and d are constants for given values of b and θ.

$$\begin{aligned} c &= |\,b\cos\theta\,| \\ d &= b\sin\theta \end{aligned}$$

If we let $c=-b\cos\theta$, then equation (6) can be used for all values of θ between 0 and 180 degrees.

Fig. 14. Model used for computing normal distance z from curvepoint (arrow) to stick, in F/B-S, as a function of ideal angle θ, given stick length s and stick placement with respect to curvepoint parameterized by distance b.

Appendix C
Alternative Definitions of Curvature

1) Let a, b and p be three points on curve Z in the sequential order a, p, b, and consider the circle C_{apb} passing through these points as a and b approach p. The limiting radius of C_{apb} is called the **radius of curvature** of Z at p. The reciprocal of the radius of curvature is the **curvature** of Z at p.

2) Let T_1 and T_2 be the tangents, and N_1 and N_2 the normals, at two neighboring points p_1 and p_2 on a curve Z. Let the point of intersection of the two normals be m. The angle between the tangents is equal to the angle between the normals: $\angle(T_1T_2) = \angle(N_1N_2)$

Let p_2 approach p_1 along the curve and consider the ratio between $\angle(N_1N_2)$ and the Euclidean distance $|p_1p_2|$ between the two curve points. In general, this limit approaches a limit, called the **curvature k** of curve Z at point p_1:

$$k=\lim_{|p_1p_2|\to 0}\frac{\angle(N_1N_2)}{|p_1p_2|}.$$

We note that k is equal to the reciprocal of the length of the line segment that is the common limit of the two segments mp_1 and mp_2:

$$\begin{aligned} k &= \lim_{|p_1p_2|\to 0}\frac{\angle(N_1N_2)}{|p_1p_2|}=\lim\frac{\sin(N_1N_2)}{|p_1p_2|} \\ &= \lim\frac{|p_1p_2|}{|mp_1||p_1p_2|}=\lim\frac{1}{|mp_1|}. \end{aligned}$$

3) Let a positive direction along a curve Z be selected, and let q, measured in the counterclockwise sense, be the angle the positive tangent to the curve makes with a fixed direction in the plane. Then the rate of change of q with respect to the

TABLE I
F/B-SALIENCY SCORES FOR A RANGE OF ANGLES AND STICK LENGTHS

Stick Length	Angle	(F/B-S) Score	Normalized Score
5	135	3.31	0.13
10	135	13.66	0.14
20	135	55.09	0.14
40	135	220.77	0.14
80	135	883.51	0.14
5	90	7.61	0.30
10	90	32.33	0.32
20	90	131.95	0.33
40	90	531.40	0.33
80	90	2130.63	0.33
5	60	9.79	0.40
10	60	43.71	0.44
20	60	183.72	0.46
40	60	752.41	0.47
80	60	3044.49	0.48
5	45	9.31	0.37
10	45	42.16	0.42
20	45	178.58	0.45
40	45	734.27	0.46
80	45	2977.01	0.47
5	30	7.34	0.29
10	30	33.52	0.34
20	30	142.59	0.36
40	30	587.52	0.37
80	30	2384.56	0.37

The values in the above table were computed using the model presented in Fig. 14, and thus they will differ somewhat from the actual scores returned by the SSS algorithm as described in Appendix B (largely because of digitization effects). The scores in the table correspond to a zero level of noise elimination.

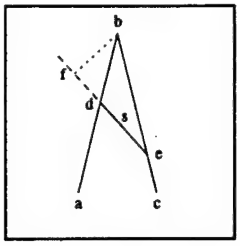


Fig. 15. Anomalous behavior of F/B-S metric for small angles. The normal distance from stick s to curvepoint b (on curve $adbec$) is bf when the stick is positioned between d and e. Intuitively, we would like the distance from b to stick s to be some quantity measured inside triangle dbe.

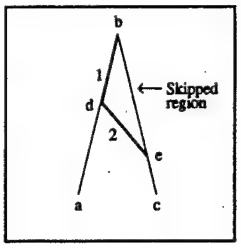


Fig. 16. Nonsymmetric scores for forward and backward movement along a curve produced by F/B-S employing a fixed Euclidean distance for stick size. Consider stick s, as shown in position 1. A slight movement clockwise will cause the stick to jump to position 2, skipping the indicated region. If the stick was scanning the curve in the reverse direction, a skipped region would occur on the opposite leg of the angle/curve.

arc length s along the curve is called the **curvature** k of the curve at the point at which it is computed.

$$k(s) = \frac{dq}{ds}.$$

Note that the fixed direction from which q is measured is arbitrary, and does not affect k; however k changes sign if we reverse the positive direction along the curve. For the case of rectangular coordinates, where the fixed direction is that of the x axis, and the curve Z is expressed as a function $y(x)$, then if

$$q = \arctan(\frac{dy}{dx}).$$

It can be shown that

$$k(x) = \frac{\frac{d^2y}{dx^2}}{[1 + (dy/dx)^2]^{\frac{3}{2}}}.$$

If z is represented in parametric form (s is arc length along the curve)

$$z = [x(s), y(s)]$$

then (see either [14] or [23])

$$k(s) = \frac{\frac{dx}{ds}\frac{d^2y}{ds^2} - \frac{d^2x}{ds^2}\frac{dy}{ds}}{[(dx/ds)^2 + (dy/ds)^2]^{\frac{3}{2}}}.$$

the above references also provide expressions for the curvature of a Gaussian convolved (smoothed) curve.

4) For discrete curves, i.e., curves represented by a list of coordinate pairs, the limiting process is not well defined. A commonly used method for approximating the curvature at a point p on curve Z in this case is to compute the angle $\angle apb$ between the two chords extending between p and points a and b on Z, where a and b are some fixed distance along the curve from p (a to the left and b to the right). If we define the curvature at p to be the reciprocal of the radius of a circle $(1/r)$ passing through the set of three points (a, p, b), then for $s = |ap|$ we have the relationship:

$$k(p) = 1/r = (2/s)\cos(\angle apb/2)$$

We note that, for angles measured between 0° and 180°, the curvature measure $1/r$ is a monotonic function of $\angle apb$. That is, the radius of curvature decreases monotonically as $\angle apb$ decreases. Thus, measuring saliency, either by fitting a circle to the curve at a point of interest or by measuring the angle between the arms of a straight-line approximation, will produce a local curvature extrema at the same location.

We also note that when $\angle(apb)$ is 180°, $k = 0$. As $\angle(APB)$ approaches 0°, r approaches $s/2$, which is geometrically correct, but $k(P)$ does not approach infinity as desired. Thus, the range of values for k is from 0 (for a straight line) to $s/2$ for an angle of 0°. Obviously, if s approaches 0, as would

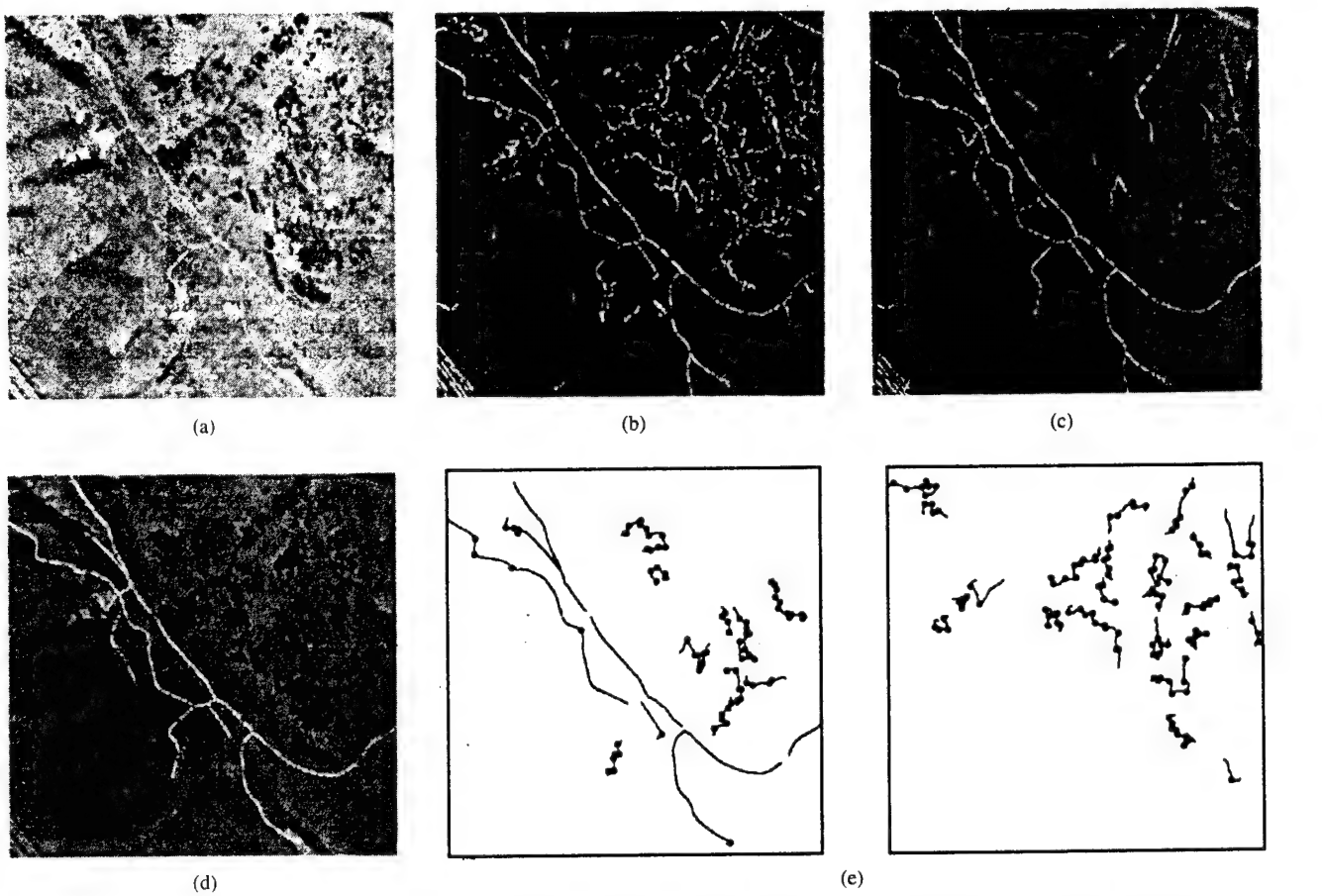


Fig. 17. Application of the SSS algorithm to the problem of delineating linear features in aerial photographs. (a) Aerial photograph. (b) Initial extraction of linear structure. (c) Filtered linear structure using SSS algorithm. (d) Delineation of major roads and trails. (e) Partition points found by SSS algorithm on curves from (b).

be the case if a limiting process could be carried out, then k would indeed approach infinity for a zero angle.

Appendix D
Performance Characteristics of the F/B-Saliency Metric

There are a number of characteristics of the basic F/B-saliency that are not immediately obvious, and could lead to anomalous behavior under appropriate conditions if not corrected for. These characteristics include the following.

- Nonmonotonic F/B-S scores if we operate over the entire range of angles between 0° and 180° (see Table I and Fig. 13). The scores increase monotonically as an ideal angle (with infinite-length sides) decreases to approximately 45°–60°; then the F/B-S scores decrease monotonically as the angle continues to decrease. This phenomenon is caused by the situation depicted in Fig. 15 and can easily be corrected for (if desired) using the modified distance metric discussed in Appendix B.
- Nonsymmetric scores for a curve segment evaluated in the forward and backward directions. This is also a consequence of the the situation depicted in Fig. 16 and is handled by evaluating the F/B-S metric in both the forward and reverse directions along the curve and adding the two scores.

Appendix E
Partitioning Curves Extracted from Aerial Imagery

A technique for detecting and delineating low-resolution linear structures appearing in aerial imagery, such as roads and rivers, was described by the authors of this paper in an earlier publication [7]. The algorithm was effective in finding such structures, but it provided no mechanism for distinguishing between the semantically meaningful objects and the accidental and irrelevant linear features found in most real images. In work now in progress, we use the SSS algorithm to slice up the individual curves found by the delineation algorithm. We throw away the very small resulting segments, which are typical of accidental linear formations, and then further filter the longer segments with respect to a set of semantic constraints. Those segments that pass through the filtering process are then "glued" back together to produce the desired delineation. This process is illustrated in Fig. 17. Fig. 17(a) shows an aerial image, and Fig. 17(b) shows the linear segments extracted by use of the original delineation algorithm. Fig. 17(c) shows those segments that passed through the filters mentioned above, and Fig. 17(d) shows the result of a final step to retain only the more significant roads and trails. The two panes of Fig. 17(e) show the results of applying the SSS algorithm to some of the 120 curves highlighted in Fig. 17(b) (they have been isolated and separated into the two panes to allow clear display

of the partition points and to prevent confusion due to the intersections of distinct curves). The robustness of the SSS algoritmn is essential in carrying out the filtering operation. Insertion of extraneous partition points would cause the loss of portions of the road network; absence of valid partition points would allow meaningless appendages to become part of the extracted network.

REFERENCES

[1] F. Attneave, "Some informational aspects of visual perception," *Psychol. Rev.*, vol. 61, pp. 183–193, 1954.

[2] A. Bengtsson and J. O. Eklundh, "Shape representation by multiscale contour approximation," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 1, pp. 85–93, Jan. 1991.

[3] A. K. Cline, "Scalar- and planar-valued curve fitting using splines under tension," *Commun. Ass. Comput Mach.*, vol. 17, no. 4, pp. 218–223, Apr. 1974.

[4] L. S. Davis, "Understanding shape: Angles and sides," *IEEE Trans. Comput.*, vol. C-26, pp. 236–242, Mar. 1977.

[5] L. Dreschler and H. Nagel, "Volumetric model and 3-D trajectory of a moving car derived from monocular TV-frame sequence of a street scene," in *Proc. 7th IJCAI*, Vancouver, Canada, pp. 692–697, Aug. 1981.

[6] M. A. Fischler and R. C. Bolles, "Perceptual organization and curve partitioning," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 8, no. 1, pp. 100–105, Jan. 1986.

[7] M. A. Fischler and H. C. Wolf, "Linear delineation," in *Proc. IEEE CVPR-83*, June 1983, pp. 351–356; also in *Readings in Computer Vision*, M. A. Fischler and O. Firschein, Eds. New York: Morgan Kaufmann, 1987, pp. 204–209.

[8] M. A. Fischler and P. Barrett, "An iconic transform for sketch completion and shape abstraction," *Comput. Graphics Image Processing*, vol. 13, pp. 334–360, 1980.

[9] D. Hilbert and S. Cohen-Vossen, "Geometry and the imagination," *Chelsea*, 1952.

[10] D. D. Hoffman and W. A. Richards, "Representing smooth plane curves for recognition: Implications for figure-ground reversal," in *Proc. 2nd Nat. Conf. Artificial Intelligence*, Pittsburg, pp. 5–8, Aug. 1982.

[11] H. Imai and M. Iri, "Computational-geometric methods for polygonal approximations of a curve," *Comput. Vision, Graphics, Image Processing*, vol. 36, no. 1, pp. 31–34, Oct. 1986.

[12] D. G. Lowe, "Organization of smooth image curves at multiple scales," in *Proc. 2nd ICCV*, pp. 558–567, 1988.

[13] R. Mehrotra, S. Nichani, and N. Ranganathan, "Corner detection," *Patt. Recogn.*, vol. 23, no. 11, pp. 1223–1233, 1990.

[14] F. Mokhtarian and A. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, no. 1, pp. 34–43, Jan 1986.

[15] T. Pavlidis and S. L. Horowitz, "Segmentation of plane curves," *IEEE Trans. Comput.*, vol. C-23, pp. 860–870, Aug. 1974.

[16] K. Rangarajan, M. Shah, and D. Van Brackle, "Optimal corner detector," in *Proc. 2nd ICCV*, Tampa, FL, Dec. 1988.

[17] W. Richards and D. Hoffman, "Codon constraints on closed 2D shapes," in *Human and Machine Vision II*, A. Rosenfeld, Ed. San Diego: Academic, 1986, pp. 207–223.

[18] W. Richards, B. Dawson, and D. Whittington, "Encoding contour shape by curvature extrema," *J. Optical Soc. Amer.*, series A, vol. 3, no. 9, pp. 1483–1491, Sept. 1986.

[19] A. Rosenfeld and E. Johnston, "Angle detection in digital curves," *IEEE Trans. Comput.*, vol. C-22, pp. 875–878, 1973.

[20] A. Rosenfeld and J. S. Weszka, "An improved method of angle detection on digital curves," *IEEE Trans. Comput.*, vol. C-24, pp. 940–941, Sept. 1975.

[21] C. H. Teh and R. T. Chin, "On the detection of dominant points on digital curves," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, no. 8, pp. 859–872, Aug. 1989.

[22] A. Witkin, "Scale space filtering," in *Proc. 8th IJCAI*, Karlsruhe, Germany, pp. 1019–1022, Aug. 1983.

[23] D. M. Wuescher and K. L. Boyer, "Robust contour decomposition using a constant curvature criterion," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 13, no. 1, pp. 41–51, Jan. 1991.

Martin A. Fischler (S'57–M'58) received the B.E.E. degree from the College of the City of New York, New York, NY, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA.

He is currently Program Director for Perception Research at the Artificial Intelligence Center of SRI International, Menlo Park, CA. He was previously a consultant and lecturer at San Jose State University, San Jose, CA, at Stanford University, and at the University of California, Berkeley.

Dr. Fischler has published two books and more than 60 research papers on computer vision, artificial intelligence, computer theory, and information theory.

Helen C. Wolf received the A.B. degree from the University of California, Berkeley, in 1958.

She is currently a Computer Scientist at the Artificial Intelligence Center of SRI International, Menlo Park, CA.

Ms. Wolf has been a coauthor of several research papers on computer vision.

Appendix B

"The Perception of Linear Structure: A Generic Linker"

ARPA Image Understanding Workshop,
Monterey, CA, November 1994

Martin A. Fischler

The Perception of Linear Structure: A Generic Linker

Martin A. Fischler *
SRI International
333 Ravenswood Avenue, Menlo Park, CA 94025, USA
(fischler@ai.sri.com)

Abstract

This paper describes the ideas, implementation details, and experimental evaluation of the linking component in a major general purpose system designed to delineate linear structures in monochrome images. It further describes the system context in which the the linker must operate and one of the generic problems it must solve. This problem, detecting a "random curve" embedded in a random noise field, is a challenging "end-to-end" problem in its own right – the paper includes an analytical and experimental evaluation of the performance of the linker in terms of its ability to duplicate human performance in this task.

1 Introduction

Figure 1 shows a block diagram of the complete system (LDS) we have designed to delineate linear structure in the context of a very wide range of applications. The first component, the detector/binarizer, accepts a graylevel image and is intended to return a binary mask that retains the linear structures of interest in a form clearly visible to a normal human observer. The second component, the generic linker (GL), uses generic criterion (continuity/contiguity, coherence/length) as the basis for extracting sequences of points that represent the perceptually obvious (curved) lines present in the binary mask. The third component, the semantic linker, uses semantic knowledge and constraints, relevant to some intended purpose, to filter and restructure the the raw output of the GL.

Even a casual examination of the delineation problem, especially in situations where semantic constraints must be invoked, shows that no single closed design can hope to be generally effective. Even to the extent that, in our system design, we have been able to separate much of the generic functionality from the semantic operations, we still have to be responsive to a significant number of distinct cases that require significantly different data structures and algorithmic techniques. In particular, we recognize (at least) four distinct types of generic linear structure: (a) open-paths, (b) closed-paths, (c) trees, and (d) networks – each of these generic types poses different computational problems, and thus, the need for different algorithmic machinery and representations. Further, the binary mask that provides the input to the GL will not be perfect – we characterize both the background and linear elements as being subject to either independent or structured noise. Thus there are at least 16 combinations of data and noise types that must be considered at the generic level. Fortunately, not all

*The research reported here was funded by the Advanced Research Projects Agency and monitored by the U.S. Army Topographic Engineering Center under contract DACA-76-92-C-0008. The author also acknowledges the many contributions made by Helen Wolf to the work presented in this paper.

of these combinations require distinct solutions, but they do impose some differences in how the system should operate to optimize performance.

In this paper we focus on the ideas and implementation details of the generic linker (GL) and describe in detail one of its modes of operation: independent noise in both the binary image (mask) background and embedded linear element(s), and a problem context of extracting a single (non-intersecting/open) linear segment. Thus, the prototype problem we must solve is:

We embed a randomly generated curve (initially, a continuous sequence of one-pixels, see Appendix 1) in a mask that has some specified ratio of one-pixels to zero-pixels. The curve, prior to embedding, has some specified percent of its one-pixels set to zero. The noise parameters and curve length are chosen so that the embedded curve is "perceptually obvious" to almost any normal human observer. We want the GL to delineate the perceived curve – i.e., the curve the human observers sees, even if it differs from the originally embedded curve.

2 Performance Evaluation

Evaluation of performance requires a specification of the "correct" answer to the problem being addressed. In our prototype problem, there are two possible correct answers: (1) "ground truth" as represented by the originally embedded curve, and (2) agreement with human perception – especially if our goal is to duplicate human performance or to interact with a human operator in the performance of some visual interpretation task. We note that because of the operation of the random processes, a particular trial could produce a situation in which the curve that best satisfies our generally valid selection criterion – and agrees with the preferences of the HVS – does not correspond to the embedded curve, and in fact, there is no principled way to recover the originally embedded curve in this case. This is the main reason for preferring agreement with human perception over ground truth. However, in order to run the large number of tests required for meaningful evaluation of performance, we must have a practical way of automatically scoring our results (Appendix 2), and comparison of the embedded and extracted curves is the only feasible alternative. Thus, the experiments described later use a combination of ground truth and visual examination for evaluation.

3 Human Performance

Our overall system design goal was to duplicate human performance in linear delineation for perceptually obvious situations. We define perceptually obvious (or "almost immediate perception," AIP) to mean a situation where the human observer appears to find the object he is looking for almost immediately, and does not alter his answer on a more detailed subsequent inspection of the image. For our prototype problem it is thus necessary to have some understanding of the relationship between background (noise) density, embedded curve density, and embedded curve length, to insure that the desired AIP condition is satisfied. There does not appear to be any relevant literature discussing this problem; although some work by Zucker [6] established the fact that a sequence of dots on a clear background will appear to be a continuous curve only if their density exceeds 0.2

Our observations in a set of both casual and formal experiments are as follows:

- We used 200x200 pixel images in all of our experiments; the images were displayed on a Symbolics LISP-machine CRT and occupied an area approximately 6 cm. on a side; some typical images are shown in Fig 2. We generated binary images with a specified uniform probability that

any given pixel would be assigned a "one" value (typically, this value ranged from .05 to 0.3). We also generated a random 2-D continuous curve (Appendix 1) which was then subjected to a process that randomly deleted points with a specified uniform probability (typically, the deletion probability ranged from 0 to 0.2). The resulting random curve was then "embedded" in the random image (the logical OR of the corresponding curve and image pixels was used to replace the original image pixel value). The human observer does not know any of the experimental parameters.

- For a given mask-noise-density, there appeared to be a threshold curve-length necessary for almost immediate perception (AIP) in most trials; but if the curve-density was too low, even an arbitrarily long curve didn't enhance the probability of AIP, in fact, the longer the curve – beyond the length necessary to nominally produce AIP – the greater the opportunity for one of the intrinsic problems (Fig 3) to occur and cause an ambiguous situation that negates AIP. If the curves were straight lines or had some other predictable shape, then this result would almost certainly have been different. It appeared to be the case that for a mask-noise-density of .1 the curve density had to be 1.0; values of mask-noise-density greater than .1 did not permit reliable AIP to occur. The minimum curve length for this case was 30 pixels as a very rough approximation. In general, for values of mask-noise-density greater than .05, the curve density had to be at least one order of magnitude (10-20 times) greater, and the minimum curve length had to monotonically increase from a minimum value of approximately 15 pixels.

4 The Algorithm

A first order solution would be to find the densest path in the image longer than some threshold length– presumably only one obvious candidate would be present. It is computationally infeasible to attempt to generate and inspect all paths, so we need effective mechanisms for (1) suggesting only those paths likely to contain the correct solution, (2) a way of selecting the best path from the restricted set of choices, and (3) a way of dealing with the residual problems presented in figures 3 and 4.

The minimum spanning tree (MST) is the best available candidate for getting the connectivity right. It is a data representation that can be efficiently computed (order $NlogN$ complexity) and which links each point to its nearest neighbor – a good start in generating dense paths. Nevertheless, the MST is not guaranteed to contain the desired solution (fig 4), and since no data representation will eliminate the intrinsic problems displayed in fig 3, it is important to eliminate as much of the noise as possible before (rather than after) we begin our search for a best path. An initial filtering step, prior to generating the final MST can be accomplished by first partitioning the image (random mask, RM) into "clusters" by a generalized form of connected component analysis ([1], [5], and Appendix 3). The key idea here is that while path-density is not a local image property, path-density is functionally related to the corresponding distribution and frequency of "gaps" in the path which can be locally detected and measured. Any curve which can be extracted from an "n-cluster" (Appendix 3) cannot have a gap of size greater than n. If we eliminate all n-clusters with fewer than k elements (for some appropriate k) we will eliminate most of the low density noise curves at the cost of removing a few small sections of the embedded curve (EC) we are searching for. (Table 6 in Appendix 4 allows us to predict the per-

centage of EC lost in the filtering process).

The above discussion provides the rationale for the following EC recovery algorithm:

1. Measure the density of the RM (assuming the embedded curve will not significantly effect this measurement) as a way of predicting both the noise statistics, and obtaining a lower bound on the density of the EC if AIP is to be possible.

2. Partition the RM into 1-clusters (Appendix 3), and "throw-away" all clusters with fewer than k pixels. The algorithm uses the measured RM density and the tables in Appendices 3 and 4 to select an appropriate value for k. For example, if the RM density is in the range 0.08 to 0.10, then EC density is assumed to be at least 0.95, and if k is set equal to 12, then on average, less than one noise cluster will be retained (Table 5) while 88% of the EC will be retained (Table 6) in approximately 6 separate 1-clusters.

3. Generate a MST for each cluster retained in the previous step and extract its diameter path [1]. "Trim" these paths to eliminate potential false tails (Fig 3) using the approach described in step (6). The result of this step (PATHS1; Fig 5) provides a framework for the final solution.

4. To recover the pieces of the EC we threw away in step (2) we now repeat steps (2) and (3) first partitioning the RM into 2-clusters. The corresponding numbers, setting k = 10, are that 100% of the EC will be recoverable, but these pieces must be extracted from approximately 100 2-clusters (Appendix 3). The result of this operation is called PATHS2 (Fig 5).

5. In order to extract the desired missing pieces of the EC from PATHS2 – which may be contaminated with a considerable amount of noise, we assign a nominal weight of 100 to each point of PATHS1 and a weight of 1 to each point of PATHS2 (we require a large differential weighting, the specific numbers are not important). We now repeat steps (2) and (3) a third time using just the weighted points in PATHS1 and PATHS2 (All-Path-Points; Fig 5). We form 20-clusters, construct the MST for each cluster, extract the highest-weight path (rather than the longest path) for each MST, and retain the highest-weight path as the desired EC. The weighting "trick" used in this step is one of the key ideas underlying the extraction algorithm; the "maxpath" algorithm [1] used to extract the diameter/longest path in a MST is able to use any attribute of the points in a branch of the MST as the criterion function to be optimized (when all the weights are set equal to 1 the longest path, in the sense of the greatest number of points, is extracted).

6. A final trimming step is invoked to (try to) eliminate false tails (Fig 3). Trimming is applied to the ends of the EC, obtained in the previous step, by first recursively deleting the end five pixels of a 15 pixel segment with density less than some threshold value based on RM density. (A conservative setting is 0.7, between the mean density of the noise produced 2-clusters and the anticipated density of the EC; see Appendix 3.) A second trimming step deletes very short end-pieces identified by the curve partitioning technique described in [4].

5 Experiments

To evaluate the the effectiveness of our system, we ran hundreds of experiments on images randomly generated as described earlier. The results of the experiments were evaluated both by direct inspection of the delineations produced by the system, and by an automatic scoring function (Appendix 2). In the formal experiments, the lengths of the random embedded curves (EC) appearing in each trial was held to the constant value of 120 pixels – a large enough value to remove modest variations in this parameter from significantly affecting the results of the experiments. We varied the random mask (RM) density from an upper value of 0.1, the point at which AIP is no longer consistently possible for human subjects regardless of EC density or curve length, to a lower value of 0.05; and we varied the EC density from an upper value of 1.0 to a lower value of 0.85 These parameter settings were not provided to the recovery algorithm.

When the EC density was sufficiently high, our system was able to perform remarkably well in recovering the actual embedded curve, and even when we embedded curves in RM's with densities of .2-.3 (in informal experiments), where humans have trouble finding the curves, we could usually perform the delineation task reasonably well (i.e., recover the originally embedded curve).

Table 1 presents the results of the formal experiments used to evaluate the competence of the algorithm described earlier. A score of 0-15 usually implies little or no visible deviation from human preference, a score of 15-30 generally implies some deviation from human preference – or an ambiguous situation with respect to where a tail should be pruned. A score greater than 30 was typically an error – both in terms of human preference and (obviously) in recovering the originally embedded curve.

The table shows that for the range of parameter values employed in the experiments, the performance of the algorithm was very good. There were no gross errors (i.e., no scores > 30), and more than 90% of the scores were in the error-free range of values. The few marginal scores could probably have been eliminated by having the algorithm use a narrower range of control settings in response to the measured RM density. On the other hand, the algorithm may require more extensive changes to deal with very low density curves embedded in low density noise fields (an underlying design assumption was that the background noise density would be high – near the upper limit of human AIP).

6 The Complete Linear Delineation System

This paper discussed one mode of operation of one of the three major functional blocks of the linear delineation system shown in Fig 1. The first block, the detector/binarizer, was described in a prior publication [3] – it work very well and has not undergone any significant changes. The third (newest) block will be described in a later publication.

The complete system (still in development) has been tested and performs well in a wide variety of potential application domains, including delineation of roads (Figs 6,7), rivers, rail-lines and other linear structures appearing in aerial imagery. It has also been used to detect terrain features for use in ground-level robotic navigation – including the skyline, ridge-lines, trees, and paths.

In normal operation, the GL is not required to employ the hierarchical strategy developed for the difficult variation of the extraction problem posed in this paper. It is usually the case that the linear structures we are interested in have long continuous segments visible in the imagery, and as is obvious from our experimental results, these segments can be

No. of Trials	RM Density	Curve Density (length 120)	Score Distribution (%)					
			0-5	6-10	11-15	16-20	21-30	> 30
50	.10	1.00	92	6	2	0	0	0
30	.08	0.95	73	17	0	10	0	0
50	.05	0.90	80	12	2	6	0	0
30	.05	0.85	50	30	14	3	3	0

Table 1: **Experimental Results**

found and extracted in a dense random noise background even if "fusion" (Appendix 3) has occurred. Actually, structured rather than random noise is the more common problem and the semantic linker was developed to handle these cases. As can be seen in Fig 1, the GL has feedback paths which allow it to automatically adjust the binarizer thresholds, both at a global level, and in local parts of the image. In "blind" applications (i.e., no human intervention) it sets the MASK density to 0.1 – this setting, which was arrived at empirically, appears to have some justification in terms of the results obtained in this paper.

7 Summary

The purpose of this paper (part of a more comprehensive document in preparation) was to describe the key ideas and algorithms comprising one of the three main functional blocks of a major system for general-purpose linear delineation being developed at SRI. Our discussion was framed in the context of solving an independently meaningful problem that is both interesting and difficult; in fact, it is a problem that challenges the human visual system. In obtaining a solution to this problem, we had to compile new data about the formation of linear structures in random noise fields – this data has general utility. In addition to the experiments related to our posed problem, we show examples of some of the more practical applications of our complete system.

8 Bibliography

1. M.A. Fischler, "Fast Algorithms for Two Maximal Distance Problems with Application to Image Analysis," Pattern Recognition, Pergamon Press, Vol 12(1), 1980, pp 35-40.

2. M.A. Fischler and O. Firschein "Association Algorithms for Digital Imagery," Twelfth Annual Asilomar Conference on Circuits, Systems, and Computers (November 1978).

3. M.A. Fischler and H.C. Wolf, "Linear Delineation," Proceedings IEEE CVPR-83, June 1983, pp 351-356; also, Readings in Computer Vision (M.A. Fischler and O. firschein, eds.), Morgan Kaufmann, pp 204-209, 1987.

4. M.A. Fischler and H.C. Wolf, "Locating perceptually salient points on planar curves," IEEE PAMI vol. 16(2):113-129, Feb. 1994.

5. A. Rosenfeld and A.C. Kak, "Digital Picture Processing," Academic Press, 1976.

6. S.W. Zucker, "Early Orientation Selection and Grouping: Type 1 and Type II Processes," McGill University Technical Report 82-6, 1982.

9 Appendices

9.1 Generation of Random Curves

1. Thirty (x,y) pairs are generated for each curve. Each value of x and y are generated by a uniform-distribution (0-1) random-number generator and then multiplied by 199 to produce numbers (coordinate-values) uniformly distributed between 0 and 199.

2. The thirty points are next linked by a minimal-spanning-tree (MST).

3. A diameter path is extracted from the MST [1], and the ordered subset of the original randomly generated points that fall along this diameter path are the input sequence provided to a spline-fitting routine which returns a continuous curve represented by a sequence of (x,y) coordinate pairs. These sequences, typically containing on the order of 300-600 points, are then subjected to a filtering process to produce the desired density – each point, considered in turn, is retained if a random number in the range 0-1 is less than the desired curve density.

4. Finally, the curves produced as described above, are trimmed from both ends to obtain the desired number of points for random curves used in our experiments.

9.2 A Scoring Function

A simple scoring function for comparing the originally embedded and recovered curves would be to count the number of elements in the (logical) exclusive-or of the two sets of points. We use a slightly more complex scoring function that takes two problems into account: (1) deletions in the originally embedded curve can be filled-in by the background noise field of the RM, and (2) diagonal sections of the originally embedded curve can be "fattened" from 8-connectivity to 4-connectivity by the background noise field of the RM. Neither of the above cases can be detected in a principled manner and should not be counted as errors. To partially compensate for these problems we permit a one-pixel displacement to be ignored in deciding if points in the two curves correspond to each other. Unfortunately, there are still some non-uniformities in the scoring that are not completely corrected: (1) lower density curves have more gaps that can be filled in a way that looks good to the eye, but are counted as errors by the scoring function; and (2) higher density noise fields tend to produce a greater amount of fattening that does not affect the visual appearance of a good match, but which again creates additional mismatches for the scoring function.

9.3 The Formation of Clusters in a (Binary) Random Noise Field

In order to configure and parameterize our delineation system to solve the problem posed in this paper, it is necessary to characterize the structure of the background noise field ("random mask," RM) and the effect the random deletion process has on the connectivity of the embedded curve (EC).

In the case of the the RM, we are primarily concerned with the formation of random curve segments of sufficient length and density to either compete with the EC, or to interact with it. Both our analysis, and algorithmic extraction technique, invoke the intermediate structure we call an n-cluster. A 1-cluster is a collection of pixels that form a single connected component (using 8 neighbor

n	RM Density
1	.445
2	.180
3	.095
4	.065
5	.035

Table 2: **Minimum values of RM density for fusion of n-clusters**

RM-Density	%Accumulation
.05	25
.10	40
.15	75
.20	125

Table 3: **1-Cluster Growth for Path-Density = 1**

RM-Density	max size of 1-clusters (1000 trials)	max size of 2-clusters (1000 trials)
.05	10	35
.10	22	162
.15	41	
.20	65	

Table 4: **Upper-limits on the size of n-clusters**

connectivity) ; an n-cluster is a generalization in which every pixel of the cluster has a neighbor in the cluster within a "checkerboard" distance [5] of n pixels: $cbdist[(x1,y1)(x2,y2)] = max[(|x1 - x2|)(|y1 - y2|)]$. Efficient methods for finding n-clusters are known [2].

The number of hierarchical levels, and the parameterization of our algorithm, depend on three properties of n-cluster formation in binary images: (1) how large can n be, for a given RM density, before (almost) all the 1-points "fuse" into a single cluster; (2) what is the distribution and upper limit on n-cluster sizes for given values of n and RM density; and (3) what is the density of a diameter-path extracted from a random n-cluster. The enclosed tables (compiled using Monte Carlo methods) answer these questions for selected values of the relevant parameters.

Table 2 shows that for the range of RM density values we are primarily concerned with in this paper, a 3-level (instead of two) hierarchical recovery strategy is possible, or for lower values of RM density, we could use 2/3-clusters instead of 1/2-clusters. Some of these options are currently being examined.

Table 3 is of interest in regard to the design of the scoring function. It shows the % increase in the number of pixels when a continuous curve is embedded in a RM of the specified density, and then extracted as a 1-cluster.

Table 4 lists the largest values of n-cluster size (and thus, also on random-noise-path length) found in 1000 trials for the specified values of n and RM density. It presents a crude summary of the design data available in the series represented by Table 5.

Table 5 represents one of the primary predictive tools used for setting the recovery algorithm parameters. It specifies the size distribution of noise 1-clusters for RM density =.1 Thus, for example, it shows that eliminating all 1-clusters with less than 12 pixels will, on average, leave less than one noise cluster to contend with in constructing PATHS1.

Similar tables for other values of RM density and n = 1,2 have also been constructed but are not shown here. For example, the table for RM density = .1 and n = 2 shows that approximately 900 2-clusters will be formed in a binary mask with density .1; 90% of the 2-clusters, containing 50% of the 1-pixels are less than 10 pixels in size. Thus, eliminating all 2-clusters containing less than 10 pixels in size (in constructing PATHS2) will still leave approximately 100 noise

Cluster Size	Prob of Membership (%)	Expected No. of clusters in 200x200 Image
1	43.4113	1736.
2	25.2907	506.
3	14.0122	187.
4	7.7993	78.
5	4.2796	34.
6	2.3565	15.71
7	1.2741	7.28
8	.7044	3.52
9	.3964	1.76
10	.2130	.85
11	.1140	.41
12	.0633	.21
13	.0370	.11
14	.0234	.07
15	.0116	.03
16	.0052	.013
17	.0043	.010
18	.0022	.005
19	.0005	.001
≥ 20	.0011	.002

Table 5: **1-cluster Statistics for RM-Density = 0.10**

clusters to examined and eliminated in searching for the EC.

The relation between a 2-cluster and its diameter-path (needed for trimming "tails" Fig 3) seems to be relatively independent of RM density for the range of RM density values between .05 and 0.1 We find that mean path density is approximately 0.65 with σ = .06 Bigger 2-clusters have a smaller density variance, and a smaller ratio of diameter path length to cluster size: approximately 0.66 for the largest clusters.

9.4 Binary Sequences with Random Deletions

The (two-level) hierarchical strategy employed by our recovery algorithm depends on filtering out most of the noise-produced line segments in the first pass (PATHS1; Fig 5) while retaining a significant percentage of the EC – preferably (on average) more than 50%. On the second pass, we want to be capable of recovering close to 100% of the EC even though a significant number of noise-produced line segments are also retained. Table 6 shows how eliminating n-clusters (n = 1,2 or gap-length = 0,1) with less than k elements (k = 8-14) will affect recovery statistics. The recovery algorithm used a length threshold (actually a minimum cluster size threshold) of between 10-12 for the formal experiments (Table 1). Note: we have not yet optimized the recovery algorithm to take full advantage of the information in the enclosed tables.

Curve Density	Length Thresh	Gap = 0 %Recover	Gap = 1 %Recover
.95	8	.94	1.00
	10	.91	1.00
	12	.88	1.00
	14	.86	1.00
.90	8	.82	1.00
	10	.74	1.00
	12	.67	.99
	14	.59	.99
.85	8	.70	.99
	10	.59	.98
	12	.47	.98
	14	.39	.97
.80	8	.50	.97
	10	.38	.96
	12	.28	.93
	14	.22	.91

Table 6: **Curve Recovery Statistics**

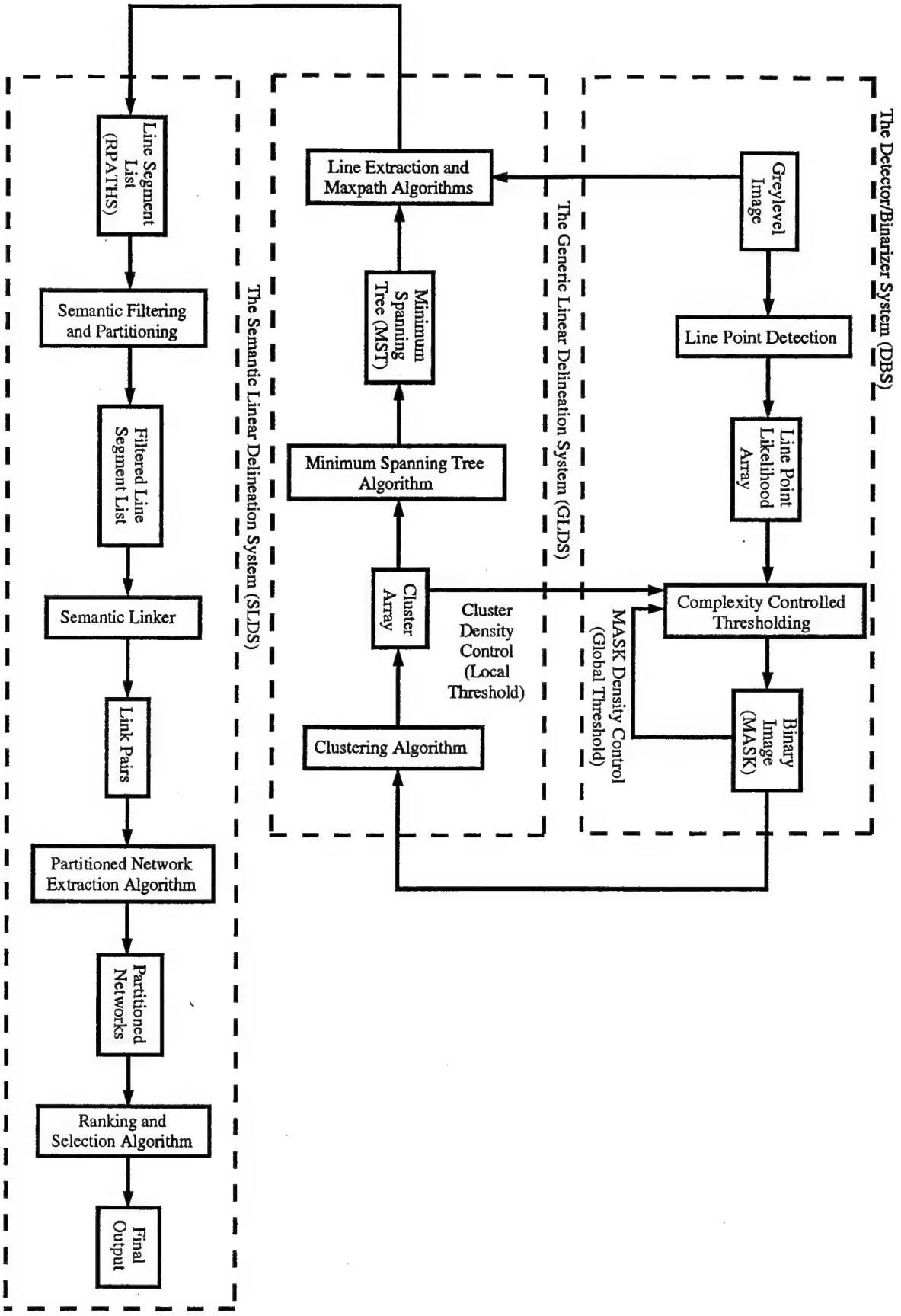


Figure 1: The Linear Delineation System (LDS)

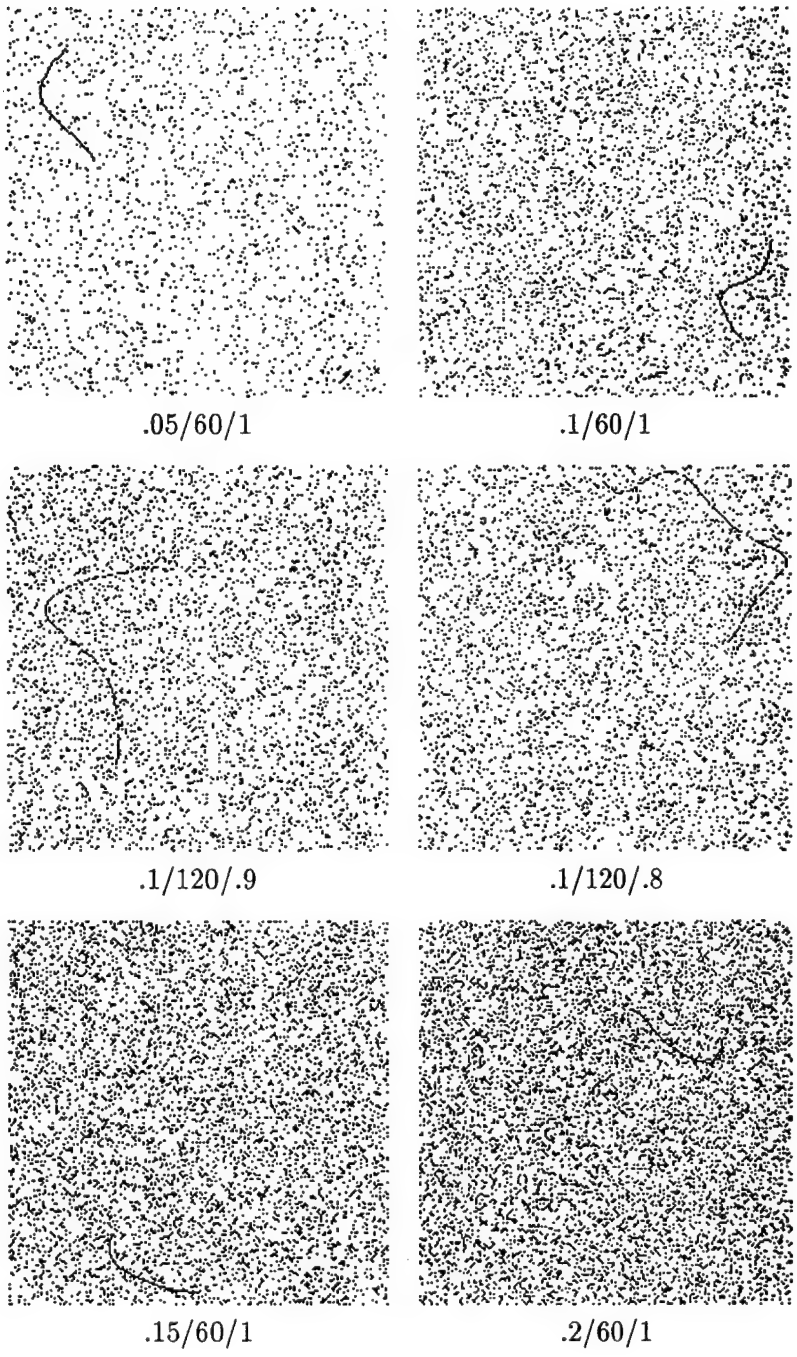


Figure 2: Example of curves embedded in a random noise field:
mask-density/curve-length/curve-density

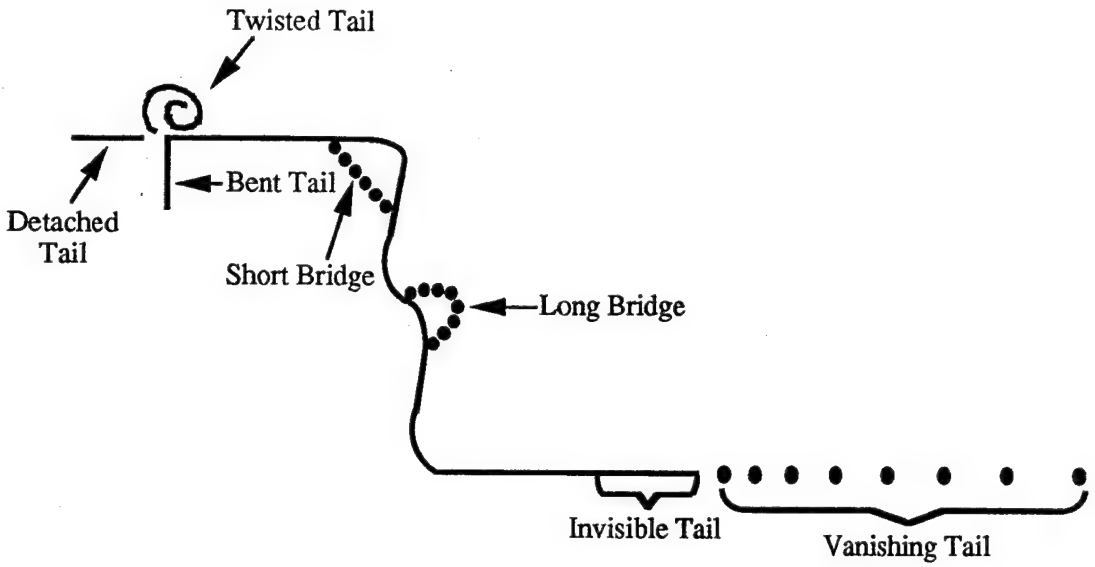


Figure 3: Intrinsic problems of line perception and delineation: bridges (which of two branches is the correct path), and tails (where does the curve end).

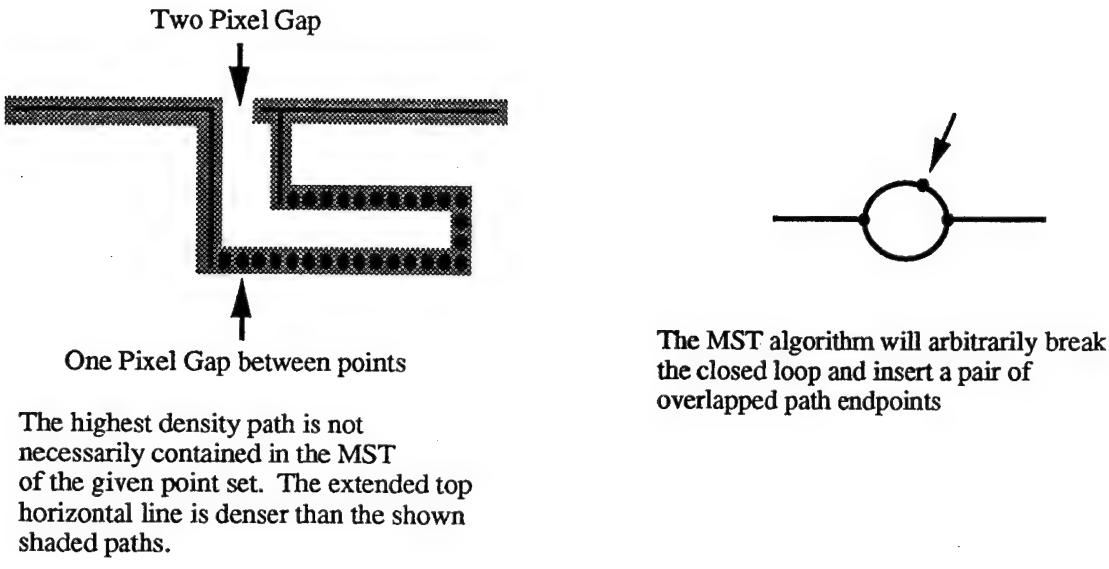
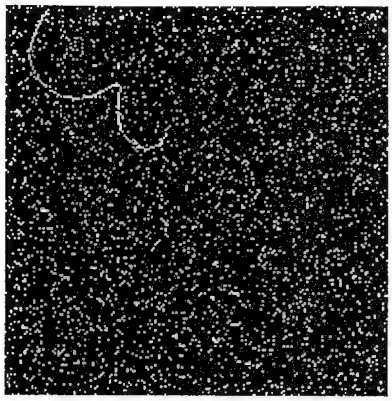
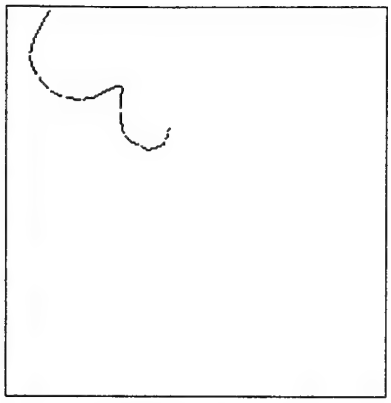


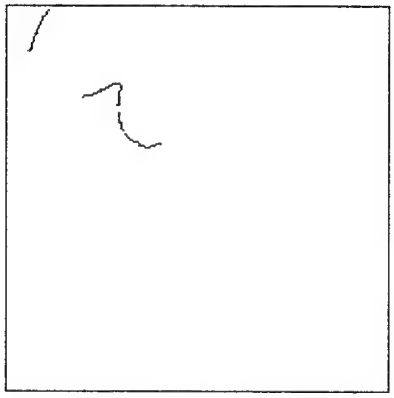
Figure 4: Minimum Spanning Tree (MST) connectivity problems.



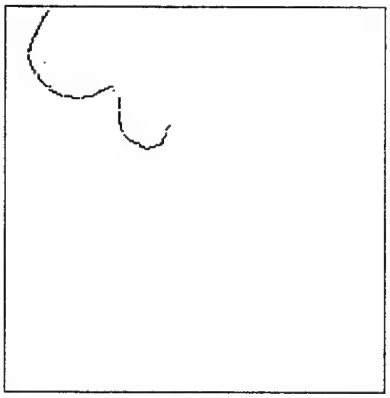
Random mask with embedded random curve.



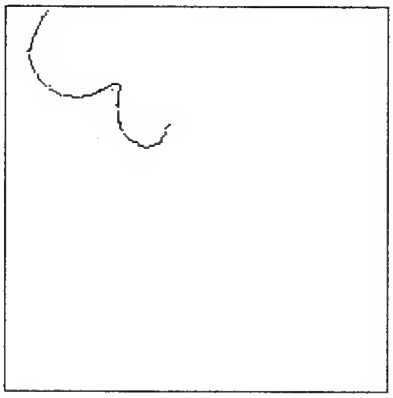
Random curve.



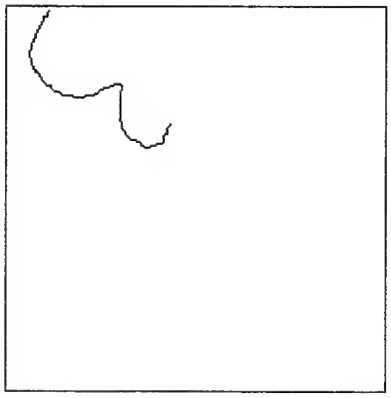
Paths 1.



Paths 2.

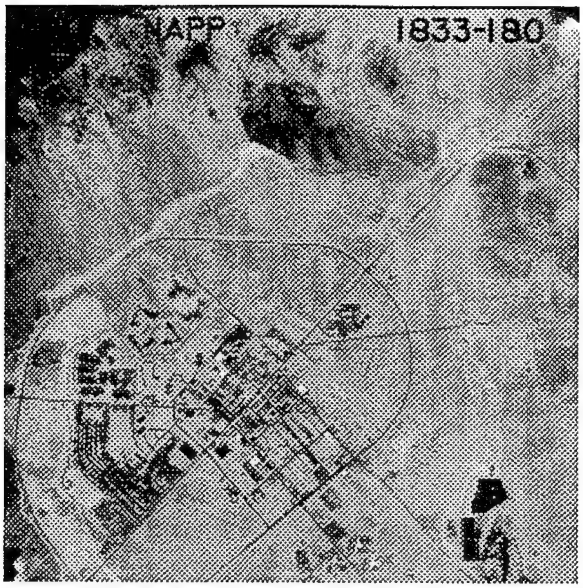


All path points.

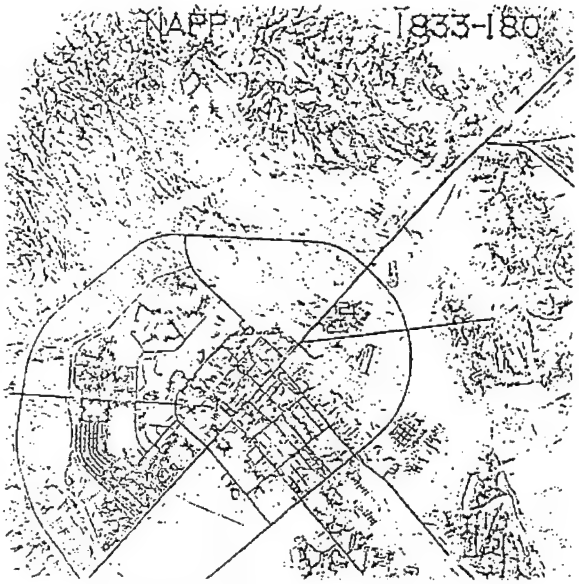


Extracted curve.

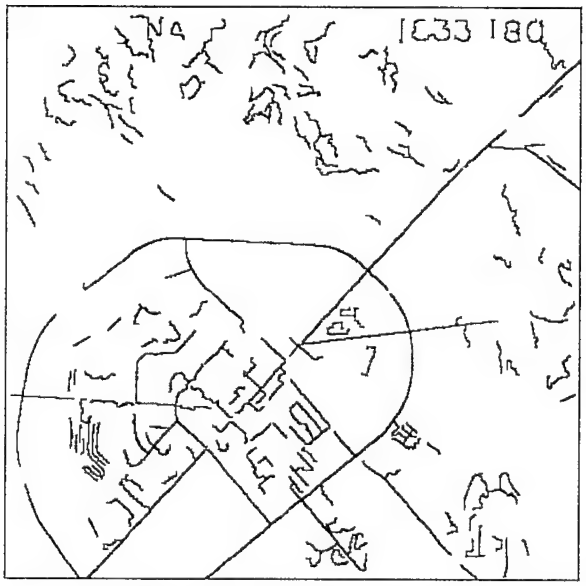
Figure 5: Example of the operation of the Generic Linear Delineation System (GLDS).



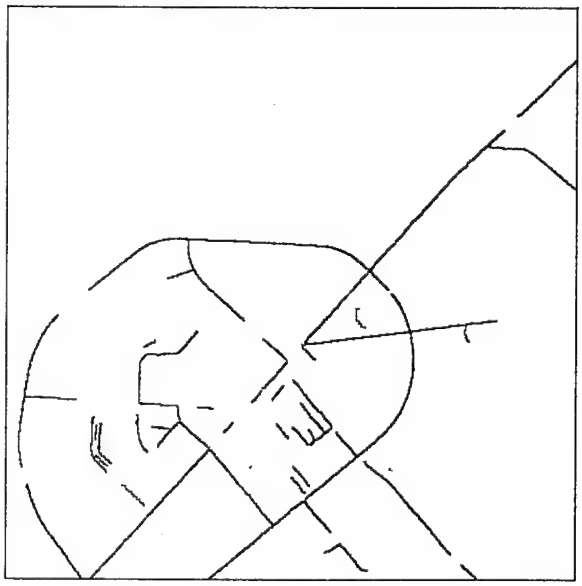
Grayscale image.



Mask of linear segment points.



Extracted set of linear segments.

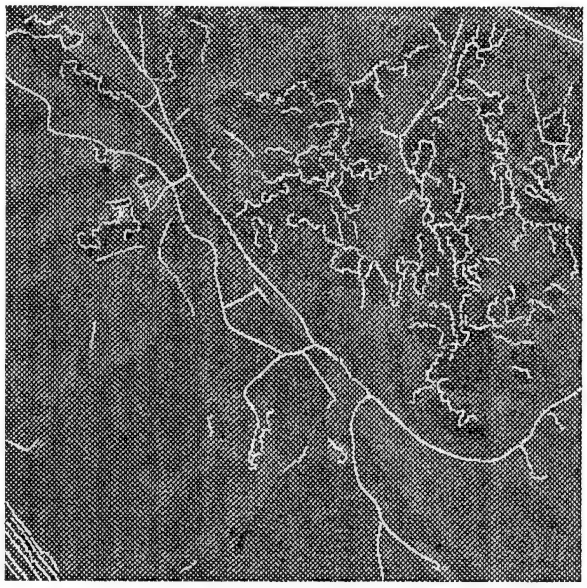


Filtered set of linear segments.

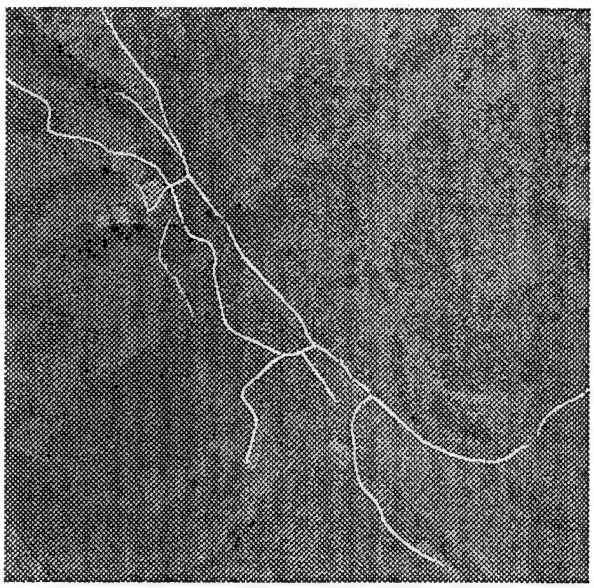
Figure 6: Example of road delineation.

Grayscale image.

Mask of linear segment points.



Extracted set of linear segments.



Filtered set of linear segments.

Figure 7: Example of road delineation.

Appendix C

"Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading"

IJCV, 1994

Pascal V. Fua and Yvan G. Leclerc

Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading

P. Fua and Y. G. Leclerc
SRI International
333 Ravenswood Avenue, Menlo Park, CA 94025
(fua@ai.sri.com leclerc@ai.sri.com)

Abstract

Our goal is to reconstruct both the shape and reflectance properties of surfaces from multiple images. We argue that an object-centered representation is most appropriate for this purpose because it naturally accommodates multiple sources of data, multiple images (including motion sequences of a rigid object), and self-occlusions. We then present a specific object-centered reconstruction method and its implementation. The method begins with an initial estimate of surface shape provided, for example, by triangulating the result of conventional stereo. The surface shape and reflectance properties are then iteratively adjusted to minimize an objective function that combines information from multiple input images. The objective function is a weighted sum of stereo, shading, and smoothness components, where the weight varies over the surface. For example, the stereo component is weighted more strongly where the surface projects onto highly textured areas in the images, and less strongly otherwise. Thus, each component has its greatest influence where its accuracy is likely to be greatest. Experimental results on both synthetic and real images are presented.

1 Introduction

The problem of recovering the shape and reflectance properties of a surface from multiple images has received considerable attention (Barrow and Tenenbaum 1978, Grimson and Huttenlocher 1992, Marr 1982, Okutomi and Kanade 1991, Terzopoulos 1988). This is a key problem not only in developing general-purpose vision systems, but also in specialized areas such as the generation of Digital Elevation Models from aerial images (Barnard 1989, Diehl and Heipke 1992, Hannah 1989, Kaiser *et al.* 1992, Wrobel 1991).

In this paper, we view the ultimate goal of a surface reconstruction method as finding an object-centered description of a surface from a set of input images that is sufficiently complete, in terms of its geometric and radiometric properties, that it is possible to generate an image of the surface from any viewpoint. In particular, the description should be sufficiently complete to reproduce the input images to within a certain tolerance, given models of the cameras, their relative locations, and expected noise.

Our surface reconstruction method uses an object-centered representation, specifically a triangulated 3–D mesh of vertices. Such a representation accommodates both geometric and radiometric information, as well as multiple images (including motion sequences of a rigid object) and self-occlusions. We have chosen to model the surface material using the Lambertian reflectance model with variable albedo. Consequently, the natural choice for the monocular information source is shading, while intensity is the natural choice for the image feature used in multi-image correspondence. Not only are these the natural choices given a Lambertian reflectance model, they are also complementary (Blake *et al.* 1985, Leclerc and Bobick 1991): intensity correlation is most accurate wherever the input images are highly textured, whereas shading is most accurate where the input images are untextured.

The reconstruction method is to minimize an objective function whose components depend on the input images and some measure of the complexity of the 3–D mesh. The method starts with an initial estimate for the mesh derived, for example, from the triangulation of conventional stereo results, and uses conjugate gradient descent to minimize the objective function. The image-dependent components of the objective function are related to the two sources of information mentioned above. We take advantage of the complementary nature of the information sources by weighting the components at each facet of the triangulated mesh according to the degree of texturing within the areas of the images that the facet projects to. The projection uses a hidden-surface algorithm to take occlusions into account.

In the following section, we describe related work and our contributions in this area. Following this we discuss some of the key issues in multi-image surface reconstruction and how to combine different sources of information for such purposes. We then describe in detail our specific procedure, discuss its behavior on synthetic data, and show some results on real images.

2 Related Work and Contributions

Three-dimensional reconstruction of visible surfaces continues to be an important goal of the computer vision research community. Initially, much of the work concentrated on $2\frac{1}{2}$–D image-centered reconstructions, such as Barrow and Tenenbaum's *Intrinsic Images* (Barrow and Tenenbaum 1978) and Marr's $2\frac{1}{2}$*–D Sketch* (Marr 1982). These view-centered surface representations have been the basis for quite successful systems for recovering shape and surface properties. Some have used single sources of information, such as sequences of range

data or intensity images (Asada *et al.* 1992, Hung *et al.* 1991), stereo (Diehl and Heipke 1992, Kaiser *et al.* 1992, Witkin *et al.* 1987, Wrobel 1991), and shading (Hartt and Carlotto 1989, Horn 1990, Terzopoulos 1988). Others have combined sources of information, such as shading and texture (Choe and Kashyap 1991), focus, vergence, stereo, and camera calibration (Abbot and Ahuja 1990). See (Aloimonos 1989) for further discussions on information fusion.

More recently, full 3–D representations have been used, such as 3–D surface meshes (Terzopoulos and Vasilescu 1991, Vemuri and Malladi 1991), parameterized surfaces (Stokely and Wu 1992, Lowe 1991), local surfaces (Ferrie *et al.* 1992, Fua and Sander 1992), particle systems (Szeliski and Tonnesen 1992), and volumetric models (Pentland 1990, Terzopoulos and Metaxas 1991, Pentland and Sclaroff 1991).

As with the methods employing $2\frac{1}{2}$–D representations, those employing 3–D representations have used a variety of single image cues for reconstruction, such as silhouettes and image features (Cohen *et al.* 1991, Delingette *et al.* 1991, Terzopoulos *et al.* 1987, Tomasi and Kanade 1992, Wang and Wang 1992), range data (Whaite and Ferrie 1991), stereo (Fua and Sander 1992), and motion (Szeliski 1991). Liedtke *et al.* (1991) first uses silhouettes to derive an initial estimate of the surface, and then uses a multi-image stereo algorithm to improve on the result. Both their approach to deriving an initial estimate for the mesh and Szeliski and Tonnesen's approach (1992) are different from ours and this is an important topic for future research.

Of special relevance to this paper is research in combining stereo and shape from shading. Using $2\frac{1}{2}$–D representations, Blake *et al.* (1985) is the earliest reference we are aware of that discusses the complementary nature of stereo and shape from shading, but meaningful experimental results are not provided. Leclerc and Bobick (1991) discuss the integration of stereo and shape from shading, but their implementation uses stereo only as an initial condition to their height–from–shading algorithm. Cryer *et al.* (1992) combine the high–frequency output of a shape from shading algorithm with the low–frequency output of a stereo algorithm using filters designed to match those in the human visual system.

Using full 3–D representations, Heipke (1992) integrates stereo and shading, but assumes that the images can be separated beforehand into zones of variable albedo (where one does stereo) and areas of constant albedo (where one does shape from shading). This is in contrast to our approach described below, in which the optimization procedure dynamically adapts to the image data.

In this paper, we unify the idea of using 3–D meshes to integrate information from multiple images with that of using multiple cues. Our specific approach to this unification has led to a number of important contributions:

- We correctly deal with occlusions by using a hidden surface algorithm during the reconstruction process.

- Our stereo technique avoids the constant depth assumption of traditional correlation-

based stereo algorithms, effectively using self-adjusting variable-sized windows in the images.

- Our approach to shape from shading is applicable to surfaces with slowly varying albedo. This is a significant advance over traditional approaches that require constant albedo.

Finally, we view the specific manner in which the multiple cues are integrated together to be an important contribution in itself. The integration is achieved by using a weighting scheme for combining shape from shading and stereo that depends on the local degree of texturing in the input images. We establish, using both synthetic and real images, that it leads to significantly better results than using either cue alone.

To demonstrate the validity of the overall approach, we have implemented a computationally effective optimization procedure, and have demonstrated that it finds good minima of the objective function on both synthetic and real images.

3 Issues in Multi-Image Surface Reconstruction

We briefly discuss here some of the key issues in multi-image surface reconstructions, and outline how we address the issues here. These outlines will be expanded upon in Section 4.

3.1 Surface Shape and Its Representation

Since the task is to reconstruct a surface from multiple images whose vantage points may be very different, we need a surface representation that can be used to generate images of the surface from arbitrary viewpoints, taking into account self-occlusion, self-shadowing, and other viewpoint-dependent effects. Clearly, a single image-centered representation is inadequate for this purpose. Instead, an object-centered surface representation is required.

Many object-centered surface representations are possible. However, practical issues are important in choosing an appropriate one. First, the representation should be general-purpose in the sense that it should be possible to represent any continuous surface, closed or open, and of arbitrary genus. Second, it should be relatively straightforward to generate an instance of a surface from standard data sets such as depth maps or clouds of points. Finally, there should be a computationally simple correspondence between the parameters specifying the surface and the actual 3-D shape of the surface, so that images of the surface can be easily generated, thereby allowing the integration of information from multiple images.

A regular 3–D triangulation is an example of a surface representation that meets the criteria stated above, and is the one we have chosen for this paper. In our implementation, all vertices except those on the edges have six neighbors and are initially regularly spaced. Such a mesh defines a surface composed of three-sided planar polygons that we call triangular

facets, or simply facets. Triangular facets are particularly easy to manipulate for image and shadow generation; consequently they are the basis for many 3-D graphics systems. These facets tend to form hexagons and can be used to construct virtually arbitrary surfaces. Finally, standard triangulation algorithms can be used to generate such a surface from noisy real data (Fua and Sander 1992, Szeliski and Tonnesen 1992).

3.2 Material Properties and Their Representation

Objects in the world are composed of many types of material, and the material type can vary across the object's surface in many ways. The key issues, therefore, are the type of material we wish to consider, and how its variation across the surface is to be represented. In general, one can represent a material type by its reflectance function, which maps the wavelength distribution and orientation of a light source, the normal to the surface, and the viewing direction into the color of the image at a point. This function is generally quite complex. However, there are reflectance functions that are not only much simpler, but are also quite common. Such functions are modeled using only one, or, at most, a few, parameters. Consequently, one can accurately model the material properties of a surface by representing these parameters at every point on the surface.

Probably the simplest, and most common, such function is the Lambertian reflectance function. For gray-level images, this function not only has a single parameter, albedo, which is the ratio of outgoing to incoming light intensity, but the image intensity is independent of viewpoint. Image intensity can therefore be used directly when computing surface properties, as explained in Section 4. For this reason, and for the time being, we have chosen to restrict ourselves to Lambertian surfaces. Possible extensions are discussed as future work in Section 6.

Having chosen a specific reflectance function, the remaining issue is how to represent the spatially varying parameter(s). In general, one needs to be able to represent independent parameter values at every point of the surface. In terms of the mesh representation of the surface, this implies some type of spatial sampling of each facet. We have chosen to use two types of spatial sampling. The first is most appropriate when the parameters vary quickly across the surface, and the second when they vary more slowly. For the former case, we use a uniform sampling of each facet, where the intersample spacing corresponds roughly to no more than one or two pixels in any of the images. For the latter case, we use a single value associated with each facet.

As we shall see later, both representations are necessary to handle the various sources of information; the relative importance of their contributions is weighted on a facet-by-facet basis as a function of the images.

3.3 Information Sources for Reconstruction

A number of information sources are available for the reconstruction of a surface and its material properties. Here, we consider two classes of information.

The first class comprises those information sources that do not require more than one image, such as texture gradients, shading, and occlusion edges. When using multiple images and a full 3-D surface representation, however, we can do certain things that cannot be done with a single image. First, the information source can be checked for consistency across all images, taking occlusions into account. Second, when the source is consistent and occlusions are taken into account, the information can be fused over all the images, thereby increasing the accuracy of the reconstruction.

The second class comprises those information sources that require at least two images, such as the triangulation of corresponding points between input images (given camera models and their relative positions). Generally speaking, this source is most useful when corresponding points can be easily identified and their image positions accurately measured. The ease and accuracy of this correspondence can vary significantly from place to place in the image set, and depends critically on the type of feature used. Consequently, whatever the type of feature used, one must be able to identify where in the images that feature provides reliable correspondences, and what accuracy one can expect.

The image feature that we have chosen for correspondence (although it is by no means the only one possible) is simply intensity, because the Lambertian reflectance model described earlier implies that the image intensity of a surface point is independent of the viewing direction. Therefore, corresponding points should have the same intensity in all images. Clearly, intensity can be a reliable feature only when the albedo varies quickly enough on the surface (and, consequently, the images are highly textured), the search space is sufficiently narrow, and the radiometry is the same in all images. Otherwise, there would be significant ambiguity in the correspondence of pixels across the images. Differences in radiometry, however, can be accommodated by first band-passing the images (Poggio *et al.* 1985, Barnard 1989).

In contrast to our approach, traditional correlation-based stereo methods use fixed-size windows in images to measure disparities, which will in general yield correct results only when the surface is parallel to the image plane. Instead, we compare the intensities as projected onto the facets of the surface. Consequently, the reconstruction can be significantly more accurate for slanted surfaces. Some correlation-based algorithms achieve similar results by using variable-shaped windows in the images. Control Data's work (Panton 1978), the Hierarchical Warp Stereo System (Quam 1984), Nishihara's real-time stereo matcher (1984), and the adaptative windows technique described in (Kanade and Okutomi 1990) are examples of such methods. However, they typically use only image-centered representations of the surface.

As for the monocular information source, we have chosen to use shading. There are a

number of reasons for this. First, we are using a Lambertian reflectance model, making shading a relatively simple source of information. Second, shading is most reliable when the albedo varies slowly across the surface, which is the natural complement to intensity correspondence, which requires quickly varying albedo. The complementary nature of these two sources should allow us to accurately recover the surface geometry and material properties for a wide variety of images.

In contrast to our approach, traditional uses of shading information assume that the albedo is constant across the entire surface, which is a major limitation when applied to real images. We overcome this limitation by improving upon a method to deal with discontinuities in albedo alluded to in the summary of (Leclerc and Bobick 1991). We compute the albedo at each facet using the normal to the facet, a light-source direction, and the average of the intensities projected onto the facet from all images. We use the local variation of this computed albedo across the surface as a measure of the correctness of the surface reconstruction. To see why albedo variation is a reasonable measure of correctness, consider the case when the albedo of the real surface is constant. When the geometry of the mesh is correct, then the computed albedo should be approximately the same as the real albedo, and hence should be approximately constant across the mesh. Thus, when the geometry is incorrect, this will generally give rise to variations in the computed albedo that we can take advantage of. Furthermore, by using a *local* variation in the computed albedo, we can deal with surfaces whose albedo is not constant, but instead varies slowly over the surface.

3.4 Combining and Using Information Sources

Simply put, our approach to surface reconstruction is to adjust the parameters of the surface (in the case of the mesh, this means the coordinates of the vertices), until the synthesized images of the surface are most consistent with the information sources described above. This approach requires a number of things. First, one must have an initial estimate of the surface. Second, one must know the light source direction, camera models, and their relative positions—we assume these are provided a priori—so that synthetic images of the surface can be generated. Third, one must have a way of quantifying what is meant by "most consistent with the information sources." Here, we use an objective function that is a linear combination of components, one for each information source, whose weights are determined on a facet-by-facet basis as a function of the images. Finally, one must have a computationally effective means of finding a surface, given the initial estimate, that is reasonably close to the best of all possible surfaces according to the objective function.

Our combined objective function has three components, two of which were mentioned above: an intensity correlation component, and an albedo variation component. A third component is a measure of the smoothness of the surface. The first two components are weighted differently at each facet as a function of the image intensities projected onto the facet, while the surface smoothness component has the same weight everywhere, but is

typically decreased as the iterations proceed.

Since the intensity correlation component depends on the differences in image intensities at a given point on a facet, it is most accurate when the images are highly textured in the areas that the facet projects to. To see this, consider the case when the images have constant intensity in the neighborhood of the projected facet: the difference in intensity will be a constant, independent of small variations in the facet's position or orientation. On the other hand, when the images are highly textured, small changes in the facet can significantly change the value of this component. Thus, we weight the intensity correlation component most strongly for those facets in which the projected image intensities are highly textured.

Conversely, the albedo variation component is most accurate when the intensities within a facet vary slowly. This is because we are assuming that the albedo varies slowly enough across the surface that a constant-albedo facet is a good model for the surface. Since the facets are planar, this should produce images whose intensities are constant within the projected facet. Thus, we weight the albedo variation component most strongly when the projected intensities within a facet vary slowly.

Since rapidly changing albedos produce highly textured image regions, our weighting scheme, in effect, turns off the shading component and turns on the stereo component in such regions. Thus, it provides the shape from shading component with boundary conditions at the edge of regions of slowly varying albedo.

The surface smoothness component is required as a stabilizing term because neither of the above components is likely to be exactly correct, the surfaces are not exactly Lambertian, and the camera positions are not exactly correct: there is noise in the images, and so on. Currently, we use the heuristic technique of starting with a relatively large weight for the smoothness component, and decrease it as the iterations proceed. The theoretically optimal point at which the smoothness weight should no longer be decreased is still an open question. Nonetheless, a single empirically determined value has been used with great success across all of the images presented in this paper when simultaneously using stereo and shape from shading.

4 Details of Surface Model and Optimization Procedure

As discussed in the previous section, our approach to recovering surface shape and reflectance properties from multiple images is to deform a 3–D representation of the surface so as to minimize an objective function. The free variables of this objective function are the coordinates of the vertices of the mesh representing the surface, and the process is started with an initial estimate of the surface. For the experiments described in this paper, we have derived this initial estimate using one of the various methods mentioned in Section 5.

The simplest one is to triangulate the smooth depth-map generated by the correlation-based stereo algorithm described in (Fua 1993).

4.1 Images and Camera Models

In this paper, we assume that images are monochrome, and that their camera models are known *a priori*. The set of gray-level images is denoted $\mathbf{G} = (g_1, g_2, \ldots, g_{n_g})$. A point in an image is denoted $\mathbf{u} = (u, v)$, and the intensity of point $\mathbf{u}$ in image g_i is denoted $g_i(\mathbf{u})$. For noninteger values of $\mathbf{u}$ we use bilinear interpolation over the four points represented by the floor and ceiling of the coordinates of $\mathbf{u}$.

The projection of an arbitrary point $\mathbf{x} = (x, y, z)$ in space into image g_i is denoted $\mathbf{m}_i(\mathbf{x})$. There are well-known methods for correcting both geometric and radiometric errors in images, as surveyed in (Baltsavias 1991). Thus, we assume that all effects of lens distortion and the like have been taken care of in producing the input images, so that the projection of a surface into an image is well modeled by a perspective projection. Thus, $\mathbf{u} = \mathbf{m}_i(\mathbf{x})$ can be written as:

$$\begin{aligned} \begin{bmatrix} U \\ V \\ W \end{bmatrix} &= M_i \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \\ u &= U/W \\ v &= V/W, \end{aligned}$$

where M_i is a three-by-four projection matrix.

4.2 Surface Representation

We represent a surface $\mathcal{S}$ by a hexagonally connected set of vertices $\mathbf{V} = (v_1, v_2, \ldots, v_{n_v})$ called a *mesh*. The position of vertex v_j is specified by its Cartesian coordinates (x_j, y_j, z_j). Each vertex in the interior of the surface has exactly six neighbors. Vertices on the edge of a surface may have anywhere from two to five neighbors.

Neighboring vertices are further organized into triangular planar surface elements called *facets*, denoted $\mathbf{F} = (f_1, f_2, \ldots, f_{n_f})$. In this work, we require that the initial estimate of the surface have facets whose sides are of equal length. The objective function described below tends to maintain this equality, but does not strictly enforce it. The representation can be extended in a straight-forward fashion to support different surface resolutions by subdividing facets (which we have done but do not describe in detail here). However, facets of a given resolution will still be required to have approximately equal sides.

4.3 Objective Function

The objective function $\mathcal{E}(\mathcal{S})$ that we use to recover the surface is best described in two equations. In the first equation,

$$\mathcal{E}(\mathcal{S}) = \lambda_D \mathcal{E}_D(\mathcal{S}) + \mathcal{E}_G(\mathcal{S}), \tag{1}$$

$\mathcal{E}(\mathcal{S})$ is decomposed into a linear combination of two components. The first component, $\mathcal{E}_D(\mathcal{S})$, is a measure of the deformation of the surface from a nominal shape, and is independent of the images. This nominal shape represents the shape that the surface would take in the absence of any information from the images. For this paper, it is a plane. Higher-order measures, such as deformation from a sphere, are also possible.

The second component,

$$\mathcal{E}_G(\mathcal{S}) = \lambda_C \mathcal{E}_C(\mathcal{S}) + \lambda_S \mathcal{E}_S(\mathcal{S}) \tag{2}$$

depends on the images, and is the one that drives the reconstruction process. It is further decomposed into a linear combination of the two information sources described in the previous section: a multi-image correlation component, $\mathcal{E}_C(\mathcal{S})$, and a component that depends on the shading of the surface, $\mathcal{E}_S(\mathcal{S})$.

These components, and their relative weights, are described in more detail below.

4.3.1 Surface Deformation Component

As stated earlier, the surface deformation (or smoothness) component is a measure of the deviation of the mesh surface from some nominal smooth shape. When the nominal shape is a plane, we can approximate this as follows.

Consider a perfectly planar hexagonal mesh for which the distances between neighboring vertices are exactly equal. Let the neighbors of a vertex v_i be ordered in clockwise fashion and let us denote them $v_{N_i(j)}$ for $1 \leq j \leq 6$. This notation is depicted in Figure 1(a). If the hexagonal mesh was perfectly planar, then the third neighbor over from the j^{th} neighbor, $v_{N_i(j+3)}$, would lie on a straight line with v_i and $v_{N_i(j)}$. Given that the intervertex distances are equal, this implies that coordinates of v_i equal the average of the coordinates of $v_{N_i(j)}$ and $v_{N_i(j+3)}$, for any j.

Given the above, we can write a measure of the deviation of the mesh from a plane as follows:

$$\mathcal{E}_D(\mathcal{S}) = \sum_{i=1}^{n_v} \sum_{\substack{j=1 \\ k=N_i(j) \\ k'=N_i(j+3)}}^{3} (2x_i - x_k - x_{k'})^2 + (2y_i - y_k - y_{k'})^2 + (2z_i - z_k - z_{k'})^2$$

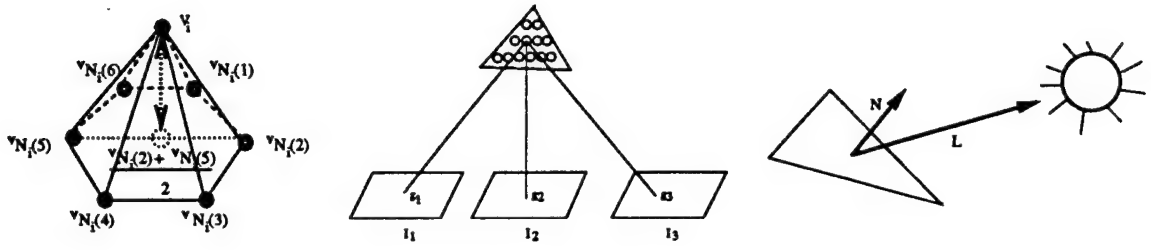


Figure 1: (a) The six neighbors $N_i(j)$ of a vertex v_i are ordered clockwise. The deformation component of the objective function tends to minimize the distance between v_i and the midpoint of diametrally opposed neighbors, represented by the dotted circle. (b) Facets are sampled at regular intervals as illustrated here. We use the gray levels of the projections of these sample points to compute the stereo score. (c) The albedo of each facet is estimated using the facet normal $\overrightarrow{N}$, the light source direction $\overrightarrow{L}$ and the average gray level of the projection of the facet into the images.

Note that this term is also equivalent to the squared directional curvature of the surface when the sides have approximately equal lengths (Kass *et al.* 1988). This term can be made to accommodate multiple resolutions of facets by normalizing each term by the nominal intervertex spacing of the facets.

4.3.2 Multi-Image Intensity Correlation

The multi-image intensity correlation component is the sum of squared differences in intensity from all the images at a given sample-point on a facet, summed over all sample-points, and summed over all facets. This component is presented in stages in the remainder of this subsection.

First, we define the sample-points of a facet by noting that all points on a triangular facet are a convex combination of its vertices. Thus, we can define the sample-points $\mathbf{x}_{k,l}$ of facet f_k as:

$$\mathbf{x}_{k,l} = \lambda_{l,1}\,\mathbf{x}_{k,1} + \lambda_{l,2}\,\mathbf{x}_{k,2} + \lambda_{l,3}\,\mathbf{x}_{k,3},\ l = 4, \ldots n_s,$$

where $\mathbf{x}_{k,1}$, $\mathbf{x}_{k,2}$, and $\mathbf{x}_{k,3}$ are the coordinates of the vertices of facet f_k, and $\lambda_{l,1}+\lambda_{l,2}+\lambda_{l,3} = 1$. In practice, $\lambda_{l,1}$ and $\lambda_{l,2}$ are both picked at regular intervals in $[0,1]$ and $\lambda_{l,3}$ is taken to be

$1 - \lambda_{l,1} - \lambda_{l,2}$. In the top half of Figure 1(b), we see an example of the sample-points of a facet.

Next, we develop the sum of squared differences in intensity from all images for a given point $\mathbf{x}$. Recall that a point $\mathbf{x}$ in space is projected into a point $\mathbf{u}$ in image g_i via the perspective transformation $\mathbf{u} = \mathbf{m}_i(\mathbf{x})$. Consequently, the sum of squared differences in intensity from all the images, $\sigma'^2(\mathbf{x})$, is defined by:

$$\begin{aligned} \mu'(\mathbf{x}) &= \frac{1}{n_i}\sum_{i=1}^{n_i} g_i(\mathbf{m}_i(\mathbf{x})) \\ \sigma'^2(\mathbf{x}) &= \frac{1}{n_i}\sum_{i=1}^{n_i} (g_i(\mathbf{m}_i(\mathbf{x})) - \mu'(\mathbf{x}))^2 \end{aligned}$$

Figure 1(b) illustrates the projection of a sample-point of a facet onto several images.

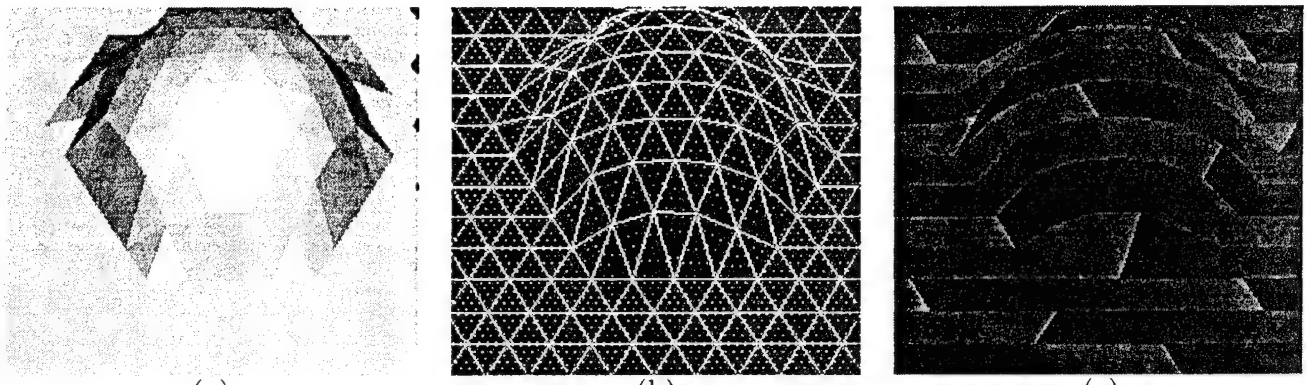
The above definition of $\sigma'^2(\mathbf{x})$ does not take into account occlusions of the surface. To do so, we use a "Facet-ID" image, shown in Figure 2. It is generated by encoding the index i of each facet f_i as a unique color, and projecting the surface into the image plane, using a standard hidden-surface algorithm. Thus, when a sample-point from facet f_k is projected into an image, the index k is compared to the index stored in the Facet-ID image at that point. If they are the same, then the sample-point is visible in that image; otherwise, it is not. Let $v_i(\mathbf{x}) = 1$ when point $\mathbf{x}$ is determined to be visible in image g_i by the method above, and $v_i(\mathbf{x}) = 0$ otherwise. Then, the correct form for the sum of squared differences in intensity at a point $\mathbf{x}$ is defined by:

$$\begin{aligned} \mu(\mathbf{x}) &= \frac{\sum_{i=1}^{n_i} v_i(\mathbf{x}) g_i(\mathbf{m}_i(\mathbf{x}))}{\sum_{i=1}^{n_i} v_i(\mathbf{x})} \\ \sigma^2(\mathbf{x}) &= \frac{\sum_{i=1}^{n_i} v_i(\mathbf{x}) \left(g_i(\mathbf{m}_i(\mathbf{x})) - \mu(\mathbf{x})\right)^2}{\sum_{i=1}^{n_i} v_i(\mathbf{x})} \end{aligned}$$

When the sample-point is visible in fewer than two images (that is, when $\sum_{i=1}^{n_i} v_i(\mathbf{x}) < 2$), the above variance has no meaning and is taken to be 0. Let s_k denote the number of facet samples for facet k for which the variance is meaningful. Summing $\sigma^2(\mathbf{x})$ over all sample-points and over all facets and normalizing by the number of meaningful sample-points yields the multi-image intensity correlation component:

$$\mathcal{E}_C(\mathcal{S}) = \frac{\sum_{k=1}^{n_f} c_k \sum_{l=4}^{n_s} \sigma^2(\mathbf{x}_{k,l})}{\sum_{k=1}^{n_f} s_k},$$

where c_k is a number between 0 and 1 that weights the contribution from each facet differently, depending on the average degree of texturing within a facet (see Section 4.3.4).



(a) (b) (c)

Figure 2: Illustration of the projection of a mesh, and the "Facet-ID" image used to accommodate occlusions during surface reconstruction. (a) A shaded image of a mesh. (b) A wire-frame representation of the mesh (bold white lines) and the sample-points in each facet (interior white points). (c) The "Facet-ID" image, wherein the color at a pixel is chosen to uniquely identify the visible facet at that point (shown here as a gray-level image).

When the original surface giving rise to the images is sufficiently textured, this component should be smallest when the surface $\mathcal{S}$ closely approximates the original surface. However, when the surface has constant, or nearly constant, albedo this component would be small for many different surfaces. As an extreme example of this ambiguity, consider a planar surface with constant albedo. This produces images with constant intensity. Thus, this component will not be able to constrain the shape of the surface, since the difference in intensity will be zero for all surfaces.

4.3.3 Shading

The shading component of the objective function is the sum, over all facets, of the difference between the computed albedo of the facet and the computed albedos of all of its neighbors. The motivation for this component, and its precise form, follow.

Recall that the Lambertian reflectance model defines the intensity g at a point on a surface with a unit surface normal $\overrightarrow{N}$ as:

$$g = \alpha(a + b\overrightarrow{N} \cdot \overrightarrow{L}), \tag{3}$$

where α is the albedo of the surface, a is the magnitude of the ambient light, b is the magnitude of a point light source, and $\overrightarrow{L}$ is the direction of the point light source as depicted

in Figure 1(c).

Note that g is independent of the viewing direction. Consequently, if we were to image a planar Lambertian facet from several points of view, its intensity would be the same for all pixels in the projection of the facet. Conversely, if we were to measure the average intensity $\bar{g}_k$ of all of the pixels within the projection of a facet f_k, we could compute its albedo, α_k, as follows:

$$\alpha_k = \frac{\bar{g}_k}{(a + b\vec{N} \cdot \vec{L})}. \tag{4}$$

This assumes, of course, that the facet is well-modeled by a single albedo, that the variation in intensity is due only to noise, and that the light source is located at infinity. In this paper, we assume that the ambient and direct illumination (*i.e.*, a, b, and $\vec{L}$) are either given or estimated from the initial surface and images, as was done in (Leclerc and Bobick 1991).

The average intensity $\bar{g}_k$ of a facet is computed by scanning over all the Facet-ID images for index k, and taking the average of the intensities at matching points in the corresponding images. This computed albedo minimizes the mean squared error between the synthesized images of the mesh surface and the input images.

Now, if the original surface had exactly constant albedo, and if our mesh surface were a good approximation to the original surface, then the computed albedos should be approximately the same across all facets. Thus, some measure of the variation in computed albedos would be a good measure of the correctness of the mesh surface. If the albedo varies slowly across the surface, we propose that an appropriate measure of this variation is the difference between the computed albedo at the facet and the computed albedos of all its neighboring facets:

$$\mathcal{E}_S(\mathcal{S}) = \sum_{k=1}^{n_f}(1 - c_k) \sum_{j \in N_f(k)} (1 - c_j)(\alpha_k - \alpha_j)^2,$$

where $N_f(k)$ is the set of indices of the facets that are neighbors of facet f_k, and c_k and c_j are numbers between 0 and 1 that depend on the degree of texturing within facets f_k and f_j.

This term can be exactly zero only where the albedo is constant. However, as will be shown in Section 5, it provides a reasonable constraint on the variation of surface normals when the albedo variation is slow. It constrains the normals of neighboring facets projecting to areas of similar gray-levels to have similar orientations. As a result, it prevents the surface normals from varying wildly in the absence of strong image gray-level variations and acts as an image-dependent regularization term that prevents the surface from wrinkling in bland areas.

4.3.4 Combining the Components

Recall that the objective function $\mathcal{E}(\mathcal{S})$ is a linear combination of three components:

$$\mathcal{E}(\mathcal{S}) = \lambda_D \mathcal{E}_D(\mathcal{S}) + \lambda_C \mathcal{E}_C(\mathcal{S}) + \lambda_S \mathcal{E}_S(\mathcal{S}),$$

where the last two components are themselves linear combinations of subcomponents computed on a per-facet basis:

$$\begin{aligned} \mathcal{E}_C(\mathcal{S}) &= (\sum_{k=1}^{n_f} c_k \sum_{l=4}^{n_s} \sigma^2(\mathbf{x}_{k,l}))/\sum_{k=1}^{n_f} s_k \\ \mathcal{E}_S(\mathcal{S}) &= \sum_{k=1}^{n_f}(1-c_k) \sum_{j \in N_f(k)} (1-c_j)(\alpha_k - \alpha_j)^2. \end{aligned} \tag{5}$$

Thus, one needs to specify both the λs, defining the relative weights of the components, and the c_ks, defining the relative weights of the two image-based components for each facet.

The λ weights are defined as follows:

$$\begin{aligned} \lambda_D &= \frac{\lambda'_D}{\| \vec{\nabla} \mathcal{E}_D(\mathcal{S}^0) \|} \\ \lambda_C &= \frac{\lambda'_C}{\| \vec{\nabla} \mathcal{E}_C(\mathcal{S}^0) \|} \\ \lambda_S &= \frac{\lambda'_S}{\| \vec{\nabla} \mathcal{E}_S(\mathcal{S}^0) \|}, \end{aligned} \tag{6}$$

where $\mathcal{S}^0$ is the initial estimate of the surface, and the λ's are user-defined weights. Normalizing each component by the magnitude of its initial gradient allows the components to have roughly the same influence when the λ's are equal. Thus, the user can more easily specify the relative contributions of each component in an image-independent fashion. This normalization scheme was used with great success in (Fua and Leclerc 1990), and is analogous to standard constrained optimization techniques in which the various constraints are scaled so that their eigenvalues have comparable magnitudes (Luenberger 1984).

As mentioned earlier, the c_k weights are a function of the degree of texturing in the intensities projected within a facet f_k. A simple measure of the degree of texturing within a facet is the variance in intensity of all the pixels projecting onto the facet, denoted $\sigma_k(\mathcal{S})$ (using the Facet-ID image to accommodate occlusions). We have empirically determined that using the logarithm of $\sigma_k(\mathcal{S})$ yields the most stable results for a large set of images:

$$c_k = a \log (1 + \sigma_k(\mathcal{S})) + b, \tag{7}$$

where a and b are normalizing factors chosen so that the smallest c_k is zero, and the largest is one.

4.4 The Optimization Procedure

The purpose of the optimization procedure is to iteratively modify the surface $\mathcal{S}$ so as to minimize $\mathcal{E}(\mathcal{S})$, given some initial estimate $\mathcal{S}^0$, and some value for the weights λ'_S, λ'_C, and λ'_D (where $\lambda'_S + \lambda'_C + \lambda'_D = 1$) defined in Equation 6. Ideally, one would like to use as small a value of the deformation weight λ'_D as possible so as to minimize the bias introduced by this term. However, in practice, λ'_D serves a dual purpose. First, since the surface deformation term is a quadratic function of the vertex coordinates, it "convexifies" the energy landscape and improves the convergence properties of the optimization procedure. Second, as discussed above and shown in Section 5, in the absence of a smoothing term, the objective function may overfit the data and wrinkle the surface excessively. Furthermore, the c_k weights of Equations 5 and 7 are computed for the initial position of the mesh and are meaningful only when it is relatively close to the actual surface.

Consequently, we use an optimization method that is inspired by the heuristic technique known as a continuation method (Terzopoulos 1986, Leclerc 1989a, Leclerc 1989b, Leclerc and Bobick 1991). We first "turn off" the shading term by setting λ'_S (Equation 6) to 0 and setting λ'_D to a value that is large enough to sufficiently convexify the energy landscape but small enough to allow curvature in the surface. In this paper, we take the initial value of both λ'_D and λ'_C to be 0.5. Given the initial estimate $\mathcal{S}^0$, a local minimum of this approximate objective function is found, using a standard optimization procedure. Then, λ'_D is decreased slightly, and the optimization procedure is applied again, starting at the local minimum found for the previous approximation. This cycle is repeated until λ'_D is decreased to the desired value. Finally we "turn on" the shading term, compute the c_k weights and reoptimize. In all examples shown in Section 5, we use $\lambda'_C = \lambda'_S = 0.4$ and $\lambda'_D = 0.2$ for this final stage.

The stereo component effectively uses only zeroth-order information about the surface (i.e., the position of the vertices), whereas shading uses first-order information about the surface (i.e., its normals). Thus, by optimizing the stereo component first, we effectively compute the zeroth-order properties of the surface and set up boundary conditions that the shading component can then use to compute the first-order properties of the surface in textureless regions. In Section 5, we will show that this leads to a significant improvement over using the stereo component alone.

When dealing with surfaces for which motion in one direction leads to more dramatic changes than motions in others, as is typically the case with the z direction in Digital Elevation Models (DEMs), we have found the following heuristic to be useful. We first fix

the x and y coordinates of vertices and adjust z alone. Once the surface has been optimized, we then allow all of the coordinates to vary simultaneously.

The optimization procedure we use at every stage is a standard conjugate-gradient descent procedure called FRPRMN (from (Press *et al.* 1986)) in conjunction with a simple line-search algorithm. The conjugate-gradient procedure requires three inputs: (1) a function that returns the value of the objective function for any $\mathcal{S}$; (2) a function that returns the gradient of $\mathcal{E}(\mathcal{S})$, that is, a vector whose elements are the partial derivatives of $\mathcal{E}(\mathcal{S})$ with respect to the vertex coordinates, evaluated at $\mathcal{S}$; and (3) an initial estimate $\mathcal{S}^0$.

Since it would be significantly slower to compute the gradient of $\mathcal{E}(\mathcal{S})$ using finite differences than analytically, we do the latter. The analytical expression of this gradient is conceptually straightforward, but is fairly complicated to derive manually. We have used the Maple [1] mathematical package to derive some of the terms. Maple directly yields the C code used in our implementation. We summarize the calculation of the derivatives below in general terms.

The derivatives of the stereo term are linear combinations of image intensity derivatives and of derivatives of the 3-D projections of points onto the images. Since we use bilinear-interpolation of image values, the first derivatives of image intensity are linear combinations of the image intensities in the immediate neighborhood of the projection. Since sample-points are linear combinations in projective space of the mesh vertices, their projections are ratios of linear combinations of the projections of the vertices, which themselves depend linearly on the vertex coordinates. Consequently, the derivatives of these projections are ratios of linear combinations of the vertex coordinates and squares of linear combinations of the vertex coordinates.

Similarly, the derivatives of the shading term depend on the derivatives of the surface normal, which can be easily derived analytically, and from the derivative of the mean gray-level in the facets. In this work, the shading term is used mainly in the fairly uniform areas where the latter derivative is assumed to be small and therefore neglected.

4.5 Computational Complexity and Convergence Issues

Each iteration of the conjugate gradient algorithm typically involves one evaluation of the gradient of the objective function and four to eight evaluations of the objective function itself. The cost of evaluating the stereo term grows as the product of the number of facets, the number of samples per facet, and the number of images. Because the albedo computation involves scanning the Facet-ID image, the dominant cost of evaluating the albedo term grows as the number of pixels per image times the number of images. The cost of evaluating the deformation term grows as the number of vertices and is small by comparison with the other two. For example, in the case of the face images shown in Subsection 5.2.2, the meshes have

[1] Trademark, Waterloo Maple Software

approximately 800 vertices and 1500 facets. We use six samples per facet and three 128x200 images. It takes about 0.6 second to evaluate the stereo energy, 0.3 second to evaluate the albedo energy and .01 second to compute the deformation energy on an R4000 SGI Indigo. Each iteration therefore takes from 5 to 10 seconds, and the computation of the final results shown in this paper took a little less than 10 minutes.

Since the optimization uses image derivatives, our technique is valid only if a majority of the facet samples project to within a few pixels of where they should be; otherwise the gradient of the objective function is meaningless and the algorithm cannot converge. This problem can be alleviated by using a coarse mesh applied to a coarse level of a gaussian pyramid, and progressively increasing the resolutions of both mesh and images. Proving the convergence of the algorithm in the general case is beyond the scope of the paper. However, in Section 5, we use both synthetic and real world examples to show that the algorithm converges when the condition stated above holds, that is, when the initial estimate is good enough for the vertices of the mesh to project to within a few pixels of their true locations.

Standard correlation-based techniques can provide starting points that have the required properties. For example, the specific algorithm we use in this paper (Fua 1993) has been shown to find few false matches and to yield a precision in the order of one pixel in disparity in the areas where it finds relatively dense matches.

5 Behavior of the Objective Function and Results

We first illustrate the behavior of the complete objective function using synthetic data. We then show that the same behavior can be observed with real data, allowing us to generate accurate 3-D reconstructions of real surfaces from multiple images.

5.1 Synthetic Data

To demonstrate the properties of the objective function of Equation 1 and the influence of the coefficients defined in Equations 6 and 7, we use as input the five synthetic images of a shaded hemisphere with variable albedo shown at the bottom of Figure 3, both with and without the addition of white noise. Each column of the figure illustrates the steps used in the creation of the image at the bottom of the column. We begin with a mesh and an albedo map, shown in the top row. Then, for each view, two images are produced. The first image (second row of the figure) is the albedo map texture-mapped onto the mesh from the final image's point of view. The second image (third row of the figure) is a shaded view of the mesh, using a constant albedo equal to one. The final image is the point-by-point product of these two images because, by Equation 3, the imaged intensity of a Lambertian surface is the product of the albedo (first image) and the inner product of the light source and the surface normal (second image).

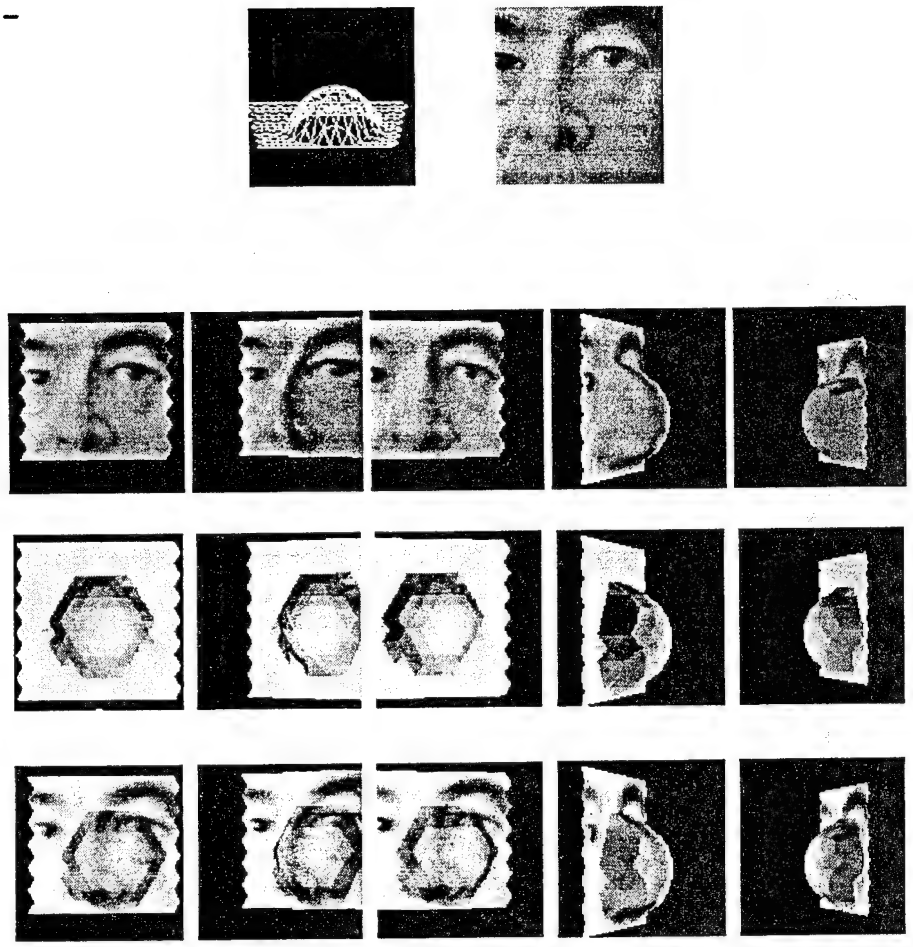


Figure 3: The making of synthetic images of a shaded hemisphere with variable albedo that conforms to our Lambertian model.

Figure 4 depicts graphically the result of our experiments. In each experiment we randomized the mesh by adding random numbers to the coordinates of the mesh vertices, and added different amounts of noise to the input images. We then used our optimization procedure to estimate the true hemispherical shape and true albedo map. More precisely, starting from our randomized initial estimate, we first use intensity correlation alone and progressively decrease the value of the λ'_D parameter of Equation 6 from 0.5 to 0. We then turn on the shading term by setting both λ'_D and λ'_S to 0.4, compute the c_ks of Equation 7, and optimize the full objective function. To show the stability of the process, we recompute the c_ks for the optimized mesh and perform a second optimization using the updated values.

The first column of Figure 4 is for experiments using only the first, second, and third im-

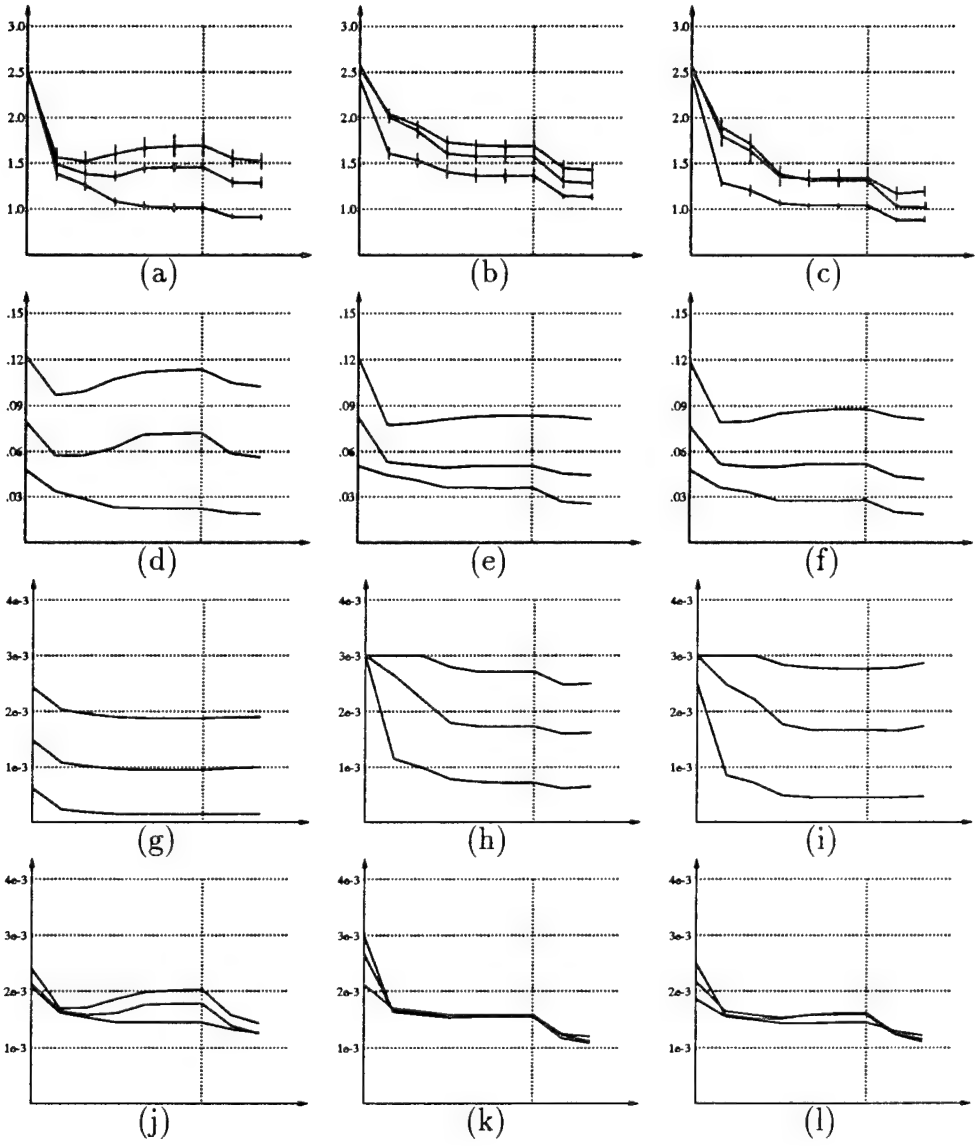


Figure 4: Graphs of the errors and objective function components while fitting a surface model to the synthetic shaded hemisphere images of Figure 3. These graphs are explained in detail in the text. (a,b,c) Average error in recovered elevation expressed in the same unit as the radius of the hemisphere, which is equal to 35. (d,e,f) Average error in recovered albedo. (g,h,i) $\mathcal{E}_C$, the stereo component of the energy. (j,k,l) $\mathcal{E}_S$, the shading component of the energy.

ages from Figure 3, where there is little self-occlusion. The second column is for experiments

using the first, fourth, and fifth images, where there is a significant amount of self-occlusion. Finally, the third column is for experiments using all five images. In this particular set of experiments, we allowed only the z coordinates of the vertices to vary. We also fixed the boundary vertices so as to eliminate the effect of the gray-level discontinuities at the border between the texture mapped part of the images and their black background .

The first row from the top of Figure 4 is a graph of the average squared error in elevation (the ordinate) versus decreasing λ'_D (the abscissa). To the left of the dotted vertical line, only the intensity correlation component is used. To the right, both the intensity correlation and shading components are used. The different curves are for different amounts of noise in the input images. The bottom curve corresponds to no noise (other than quantization error), the middle curve is for a noise variance of 4% of the image dynamic range, and the top curve is for a noise variance of 8%. The short vertical lines along the curves indicate the standard deviation of the average error over the 20 experiments performed to derive each curve.

Note that an error of 1 unit in elevation corresponds to a difference in computed disparities of approximately 0.25 pixels for projections from image 1 into images 2 or 3 and of approximately 0.80 pixels for projections from image 1 into images 4 or 5. In these experiments, the noise added to the elevations was gaussian of variance randomly chosen between 1 and 5, resulting in errors in the projections in the order of one pixel for images 2 and 3, and of three pixels for images 4 and 5. In this particular case, however, much larger errors can be tolerated: the hemisphere has a radius of 35 elevation units and can be recovered starting from a flat sheet.

The second row of Figure 4 is a graph of the average error in computed albedo. The third row is the average value of the intensity correlation component, $\mathcal{E}_C(\mathcal{S})$, and the fourth row is the average value of the shading component, $\mathcal{E}_S(\mathcal{S})$.

Note that, as λ'_D decreases and stereo alone is used (*i.e.*, as the abscissa is traversed rightwards to the dotted vertical line), the average elevation error decreases when there is no noise in the input image (bottom curve), as does the average albedo error and the two components of the objective function. However, when the images are noisy, the elevation error (first row) stops decreasing and may even begin to increase as we start fitting to the gray-level noise, even though the value of the intensity correlation component (third row) continues to decrease, as it must. Furthermore, both the albedo error (second row) and the shading component (fourth row) also begin to increase when the elevation error does. This is natural since for smaller values of λ'_D the surface becomes rougher and its normals less well-behaved. As a result, the estimated albedos of Equation 4 become less reliable and noisier.

In other words, an increase in the shading component provides us with a warning that we are starting to overfit the data. This is a valuable behavior in itself. Furthermore, by turning on the shading component of our objective function (those parts of the graphs that are to the right of the vertical dotted line), we can bring down both the error in albedo

and the value of the albedo component with at worst a modest increase in the value of the stereo component, resulting in an overall reduction of the elevation error. Even when there is nothing but quantization noise in the image, the addition of the shading component can make a small, but still noticeable difference. The reason for this is twofold:

1. The shading component averages over whole facets and is therefore less sensitive to uncorrelated noise.

2. The shading component uses absolute intensity values, whereas the stereo component uses intensity differences. Thus, in the presence of noise in textureless areas, the signal-to-noise ratio for the absolute values (used by the shading component) is larger than for the differences (used by the stereo component), thereby making the shading term more robust.

However, in our experience, the shading term can be used reliably only when the surface is relatively close to the correct answer. This is not surprising since stereo deals directly with elevations, whereas shading deals with derivatives of elevation. Consequently, we have chosen the optimization schedule described above where we first optimize using stereo alone and turn on shading only later.

There is another important point to note about these results. The elevation errors in the second column, that is, those generated using images 1, 4, and 5 with a lot of self-occlusion are very close to those of the first column, that is those generated using images 1, 2, and 3 with little self-occlusion, while those in the final column (using all five images) are significantly better. In addition, the results for images 1,4, and 5 are even slightly better than those for images 1,2, and 3 in the presence of noise because the former correspond to larger baselines. In other words, having the same number of images, but with significant self-occlusions, does not hurt our procedure. Furthermore, adding new images that contain significant self-occlusions actually improves the results.

To further demonstrate the importance of being able to combine stereo and shape from shading, even in the presence of slowly varying albedo, we present in Figure 5 a second synthetic example. If we band-pass the images using a difference of gaussians, there is not enough texture for stereo to work effectively and the surface computed using stereo alone is not very good. However by combining shape-from-shading with stereo, the result improves markedly and the recovered surface becomes very close to the synthetic one used to generate the images. In this case our starting point was a flat plane, corresponding to errors of up to 4 pixels in the initial projections of the mesh vertices.

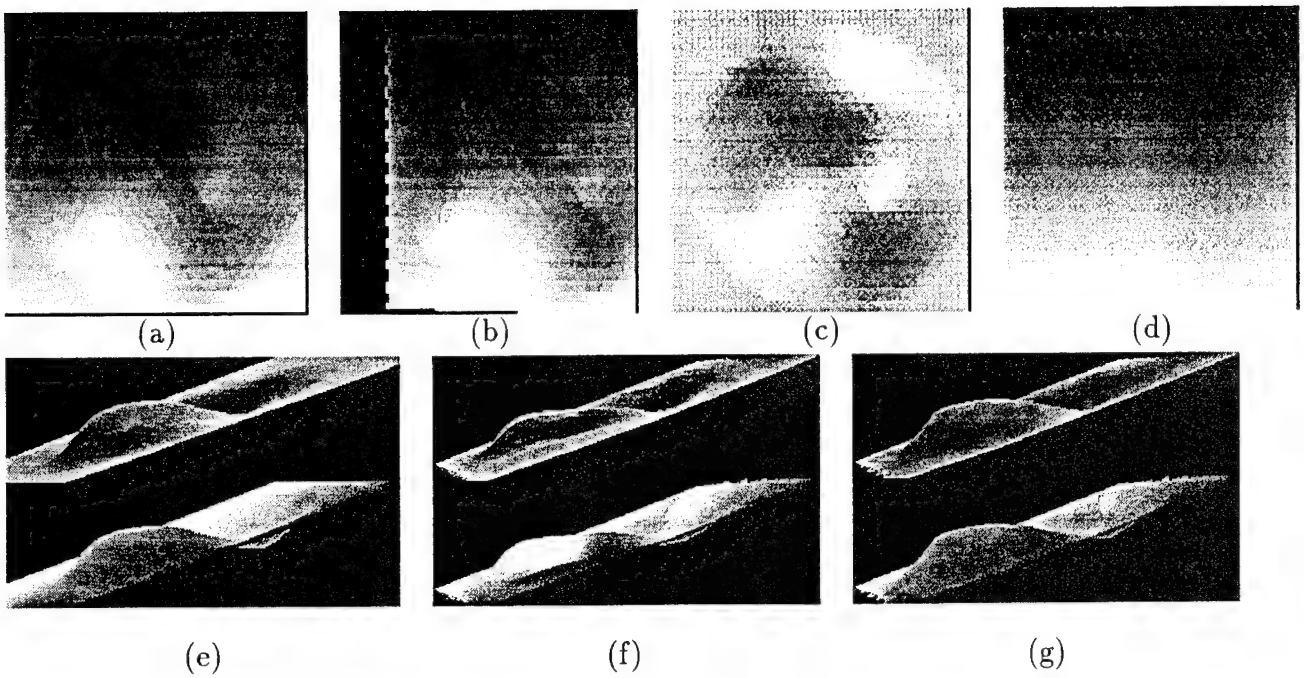


Figure 5: Combining shape from shading and stereo in the presence of a slowly varying albedo. (a,b) A synthetic stereo pair generated by rendering the shaded surface shown in (c) using the albedo map shown in (d). (e) The original shaded surface and albedo map from (c) and (d) seen from the side. (f) The surface and albedo map computed using stereo alone on difference of gaussians of the images and starting from a plane. (g) The surface and albedo map recovered by combining shape from shading and stereo. The difference-of-gaussian images do not retain enough information and stereo alone finds a poor quality solution. The shape-from-shading term, however, allows a better recovery of the surface even though the albedo is not constant.

5.2 Real Images

We now turn to real images and show that the same properties can be observed there.

5.2.1 Aerial Images

In Figure 6 we show the result of running the stereo component of our objective function on an aerial stereo pair of a sharp ridge. Note that the radiometry of the left and right images is actually slightly different. As suggested in Section 3.3, we correct for this in the computation

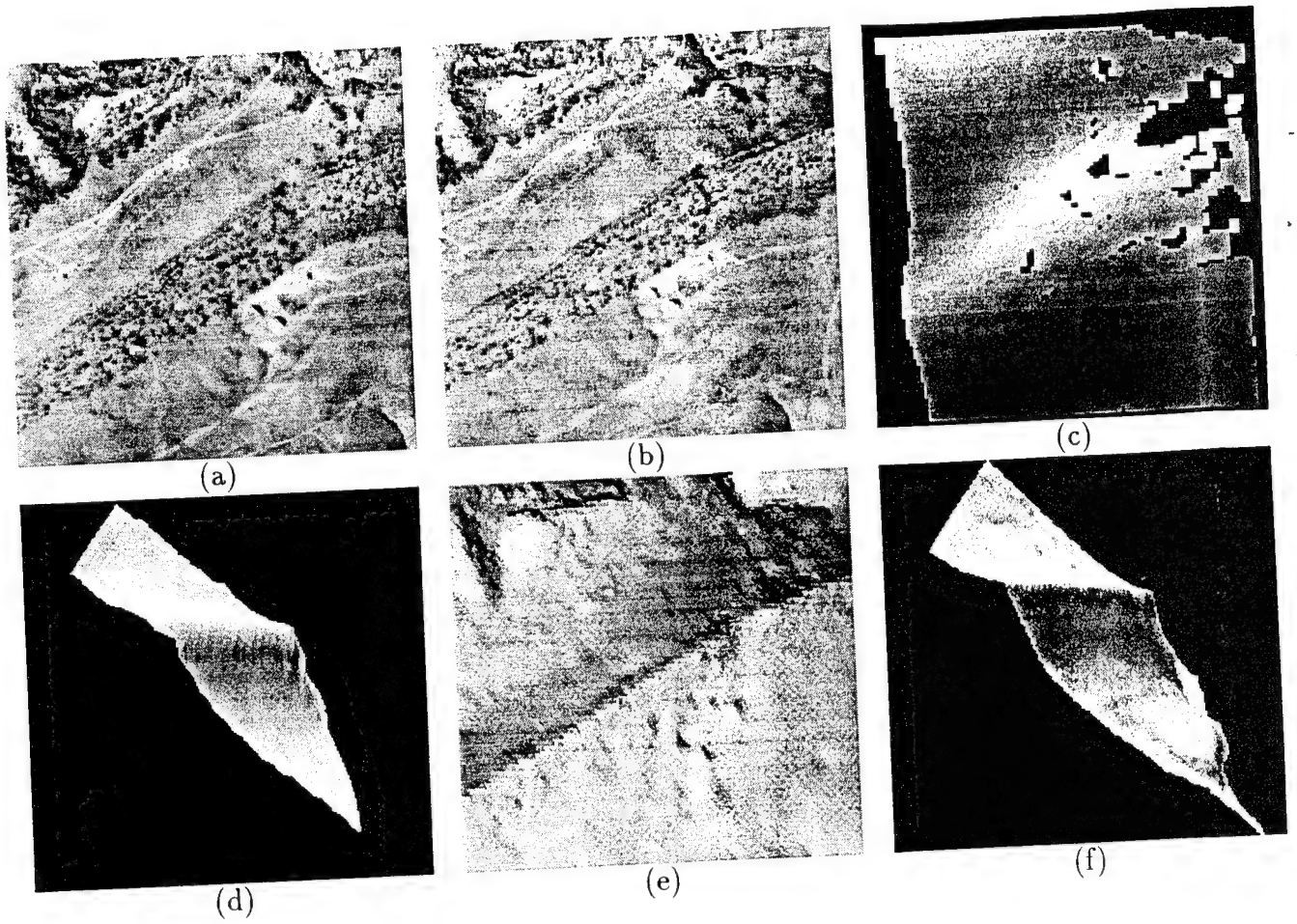


Figure 6: (a,b) A stereo pair of images of the Martin-Marietta ALV test site. (c) Disparity map computed using a correlation-based algorithm. The black areas indicate that the stereo algorithm could not find a match. Elsewhere, lighter grays indicate higher elevations. (d) The initial surface estimate derived by smoothing and interpolation of the disparity map. It is shown as a shaded surface viewed by an observer located above the upper left corner of the scene. (e,f) Shaded views of the mesh after optimization. Note that the ridge has become very sharp and that the shadow casting cliffs visible in the top portion of the image are recovered. They are clearly visible at the top of (e) and the bottom right corner of (f).

of the stereo term of our objective function by first high-pass filtering each image. Here, we use the difference between the image and its gaussian convolution.

We then optimize the mesh using the continuation method schedule described in section 4.4, that is, starting with $\lambda'_D = \lambda'_C = 0.5, \lambda'_S = 0.0$ and then progressively reducing λ'_D to 0.3 and increasing λ'_C to 0.7. Note that the recovered ridge is much sharper than in the original stereo result and that details in the upper part of the image are well recovered. The difference in the ridge elevation in the original and final estimate is approximately 40 feet, which translates to 2 pixels in disparity. Turning on the shape-from-shading term yields a result that is visually indistinguishable from the one shown here: the images are textured enough for stereo alone to be effective.

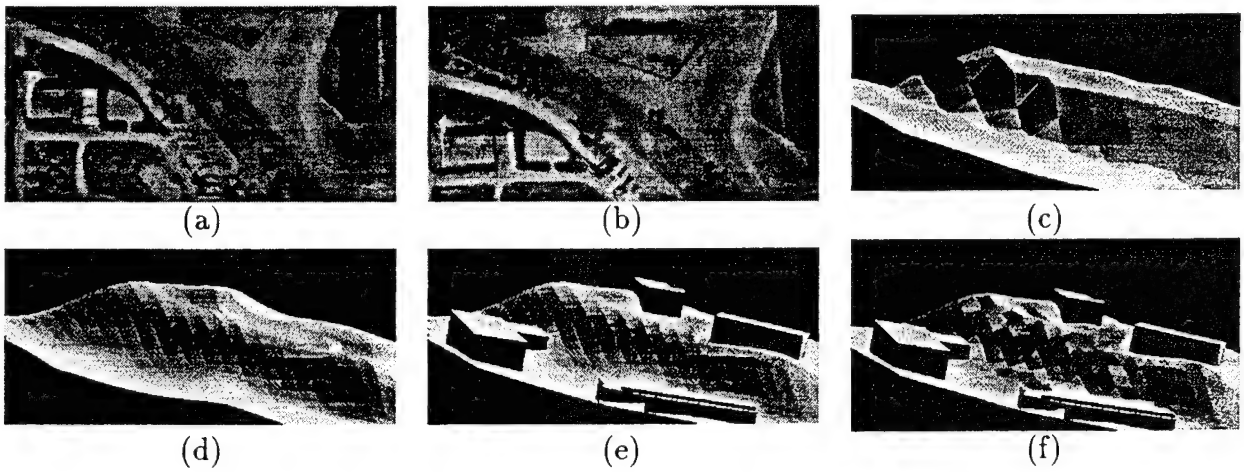


Figure 7: (a) (b) Two of a series of images of a semi-urban site. The images were taken with different light source directions. (c) A rough estimate of the ground-level surface (d) Surface after optimization using stereo alone. (e) Surface after optimization using both stereo and hand-entered buildings to mask occluded areas. (e) Surface after optimization using both stereo and shape from shading. In this case, the illuminations of the different images are different and there are very few bland, untextured areas. As a result, stereo alone performs better.

In Figure 7, we demonstrate a possible application of our technique to semiautomated cartography in a semiurban environment. We use images of a model board, each of them being taken with a different light-source direction. By fitting 3–D snakes to some of the roads in the scene and fitting a surface to them, we have generated a rough terrain model that we have then optimized using the same schedule as before. Because of the presence of buildings that cannot be well described by our mesh model and even though we use relatively large facets, the resulting surface model is too bumpy. We can improve upon this situation by manually specifying the locations of the buildings, modeling them as extruded objects, and using them to mask out occluded areas during the optimization. For comparison's sake,

we also show the result of simultaneously using stereo and shape from shading. In this case, because the illumination in each image is different and there are very few bland areas, the shape-from-shading term actually degrades the result.

5.2.2 Face Images

In Figure 8 we show two triplets of images of faces. They have been produced using the INRIA three-camera system (Faugeras and Toscani 1986) that provides us with the camera models we need to perform our computations. In this case it is essential to have more than two images to be able to reconstruct both sides of the face because of self-occlusions. For each triplet, we have computed disparity maps corresponding to images 1 and 2 and to images 1 and 3 and combined them to produce the depth maps shown in the rightmost column of the figure using the algorithms described in (Fua 1993).

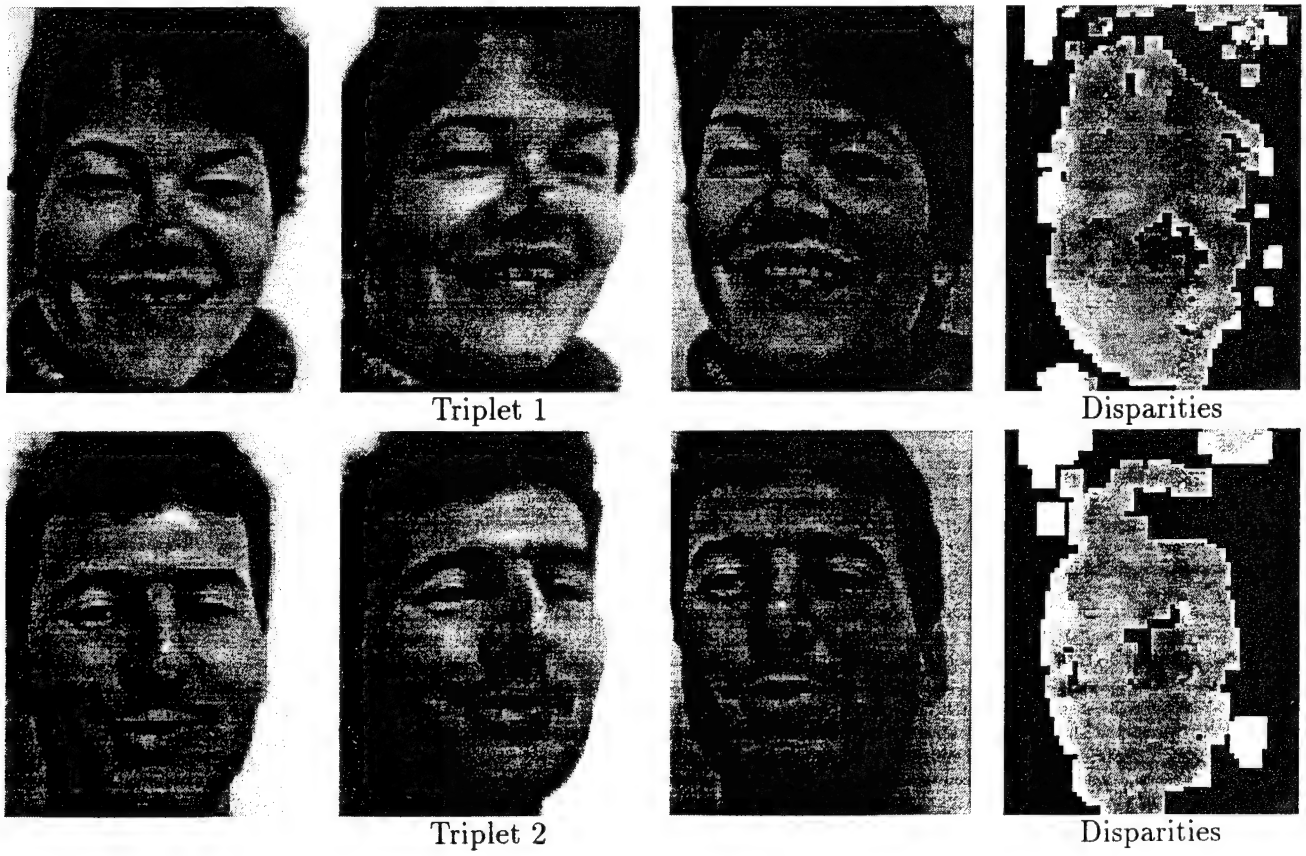


Figure 8: Triplets of face images and corresponding disparity maps (courtesy of INRIA).

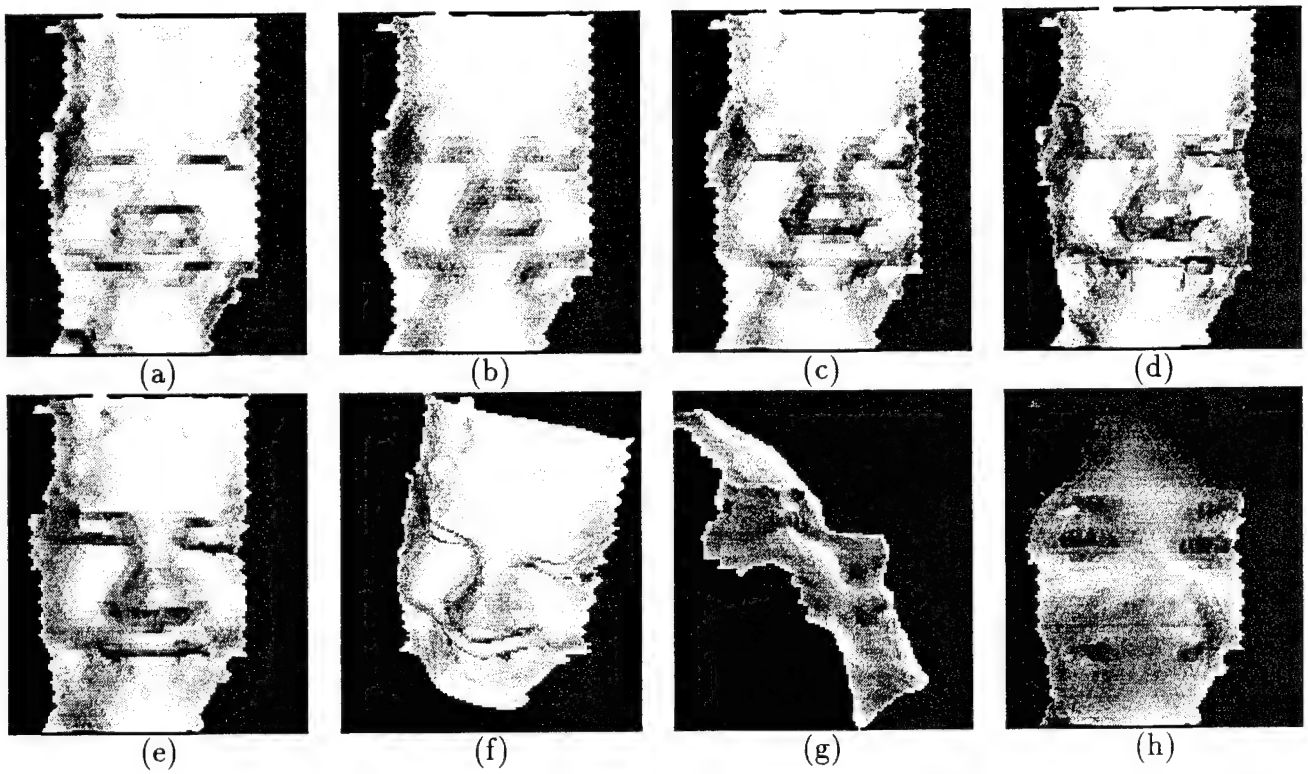


Figure 9: Results for the first triplet of Figure 8. (a) Shaded view of the mesh generated by smoothing and triangulating the computed disparity map. We use it as the starting condition for our optimization procedure. (b,c,d) The mesh after optimization using only the stereo term, with progressively less smoothing. (e,f,g) Several views of the mesh after optimization using both stereo and shading. (h) The recovered albedo map. The albedo of the nose appears fairly similar to that of the other skin areas, showing that its geometry has been well recovered. The main problem with this map is the dark streak caused by the self-shadowing crease on the right side of the face. Our algorithm does not currently handle shadows and incorrectly models them as areas of lower albedo.

The depth maps have then been smoothed and triangulated to produce the initial surfaces shown in the upper left corner of Figures 9 and 10. In the first row of these two figures, we show the result of the optimization using stereo alone as we progressively decrease the smoothness constraint and allow all three vertex coordinates to be adjusted. Note that in the first triplet (Figure 9), we recover more and more detail until the surface eventually

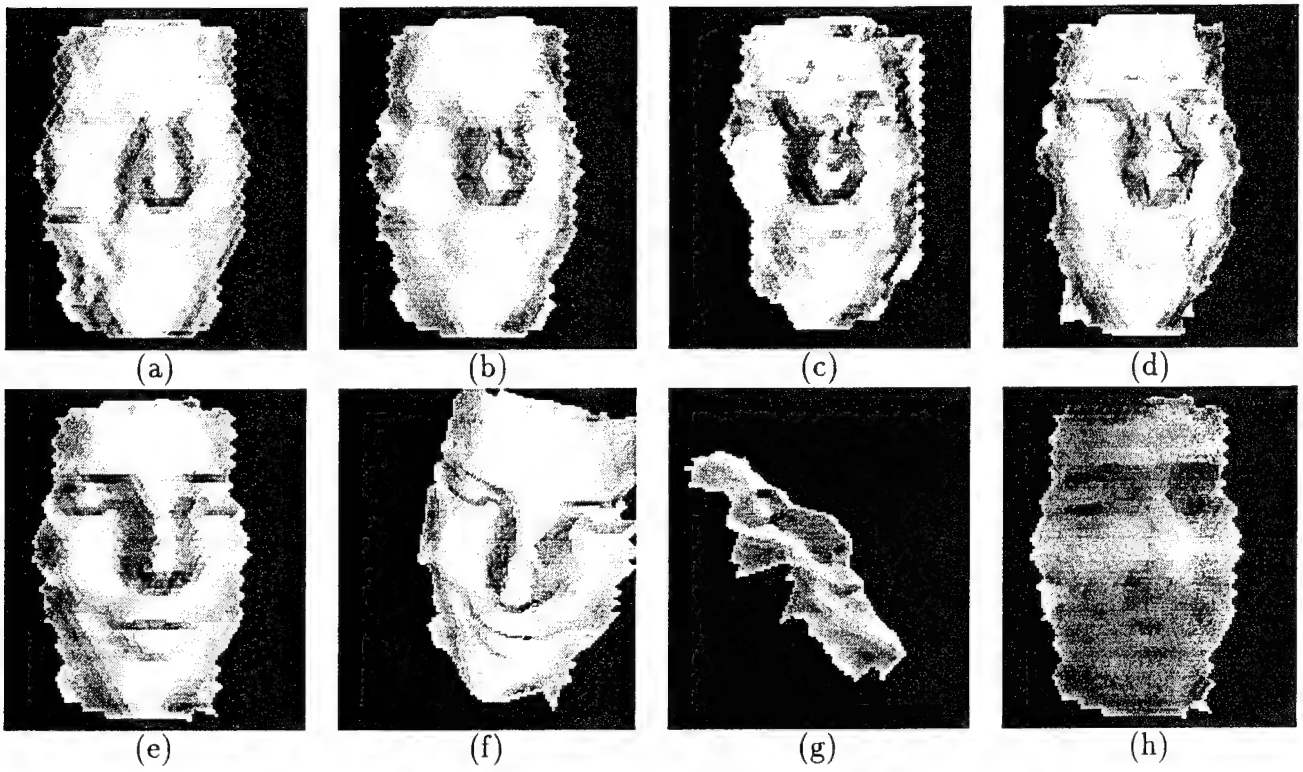


Figure 10: Results for the second triplet of Figure 8 presented in the same fashion as in Figure 9. There are strong specularities on the nose and the stereo term alone performs poorly. However, by using the shading term, the algorithm takes advantage of the monocular information present around the specularities and yields a much better result.

starts to wrinkle, without apparent improvement in accuracy. The second triplet poses an even more difficult problem: there are strong specularities on both the forehead and the nose that strongly violate our Lambertian model. Because there are very few other points that can be matched on the nose, the algorithm latches on to these specularities and yields a poor result. These two sets of images therefore present our algorithm with problems that are very similar to those discussed in Section 5.1.

In the bottom row of Figures 9 and 10, we show our final results obtained by turning on the shading term and reoptimizing the meshes. In Figure 12, we show the corresponding values of the c_k coefficients of Equation 7 and the contribution of each facet to the overall energy. For these images we did not know *a priori* the light-source direction; we therefore estimated it by choosing the direction that minimized the shading component of the objective

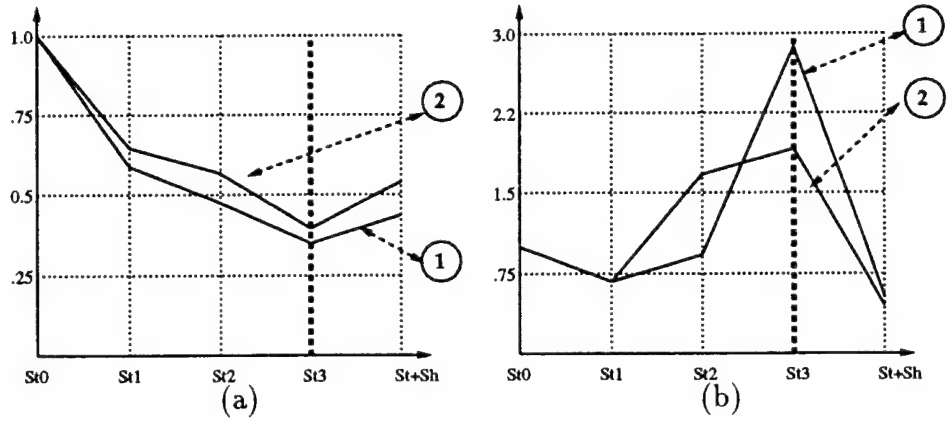
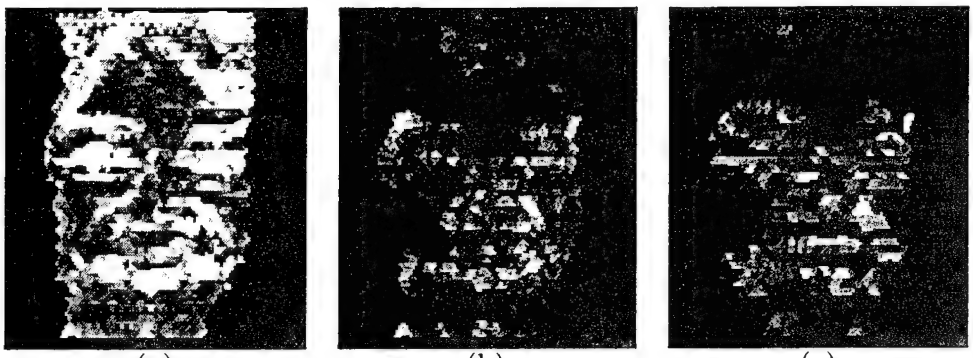


Figure 11: Values of the stereo (a) and shading (b) components of the objective function for the face images. The y axis represents the value of the components, and the x axis represents the various stages of the optimization. From left to right, we first use only stereo and decrease the smoothness and, to the right of the thick dotted line, we turn on the shading term. Each curve is labeled with the number of the corresponding image triplet, and all values have been scaled so that the initial ones are equal to 1.0.

function given the surface optimized using only the stereo component. The main features of both faces—nose, mouth, and eyes—have been correctly recovered. The improvement is particularly striking in the case of the face in Figure 10. The shading component was able to achieve this result because it uses the monocular information around the specularities. The stereo component cannot take advantage of the information around the specularities because very few points are visible in at least two images simultaneously, and because there is little texture. Of course, the effect of the specularities has not completely disappeared (there is indeed still a small artifact on the nose), but has been outweighed by the surrounding information. A more principled approach to solving this problem would be to explicitly include a specularity term in our shading model.

The graphs of Figure 11 depict the behavior of the stereo and shading components of the objective function for the two triplets. The four values of the scores to the left of the thick dotted line, St_0 to St_3, correspond to the results shown in the top row of Figures 9 and 10. The fifth value, $St + Sh$, corresponds to the final results when shading is turned on. These values have been scaled so that St_0 is equal to one for both triplets. As in the synthetic case, when using stereo alone, the stereo component always improves, but as the recovered surface becomes rougher the shading term degrades dramatically, indicating excessive wrinkling of the surface. However, when we turn on the shading component, the overall results improve



(a) (b) (c)

Figure 12: (a) Values of the c_k coefficients of Equation 7 for the final face result of Figure 9. The stereo term is dominant in areas where albedo changes rapidly such as the mouth and the eyes, and the shading term elsewhere. (b) The contribution of each facet to the stereo term. (c) The contribution of each facet to the shape from shading term.

significantly, even though the stereo component degrades slightly. For both faces, the major differences between the initial and final estimates occur in the nose area. In the original meshes, the nose tends to be oversmoothed resulting in differences of up to 15mm in terms of distance to the camera planes, or approximately six pixels in terms of disparity.

5.2.3 Ground-level Scene

In our final example, shown in Figure 13, we reconstruct a ground-level scene using three triplets of images acquired by the INRIA mobile robot. For each triplet, we have computed a correlation map. We have then used the technique described in (Fua and Sander 1992) to merge the resulting 3–D points and generate a Delaunay triangulation. Because the tops of some of the rocks are sharply slanted, the result is relatively rough and can be refined using our technique. As before, stereo alone with little smoothing yields a surface that is too wrinkly. However, by combining stereo and shape from shading, we compute a surface model in which the rock silhouettes are well defined.

6 Summary and Conclusion

We have presented a surface reconstruction method that uses an object-centered representation (a triangulated mesh) to recover geometry and reflectance properties from multiple images. It allows us to handle self-occlusions while merging information from several view-

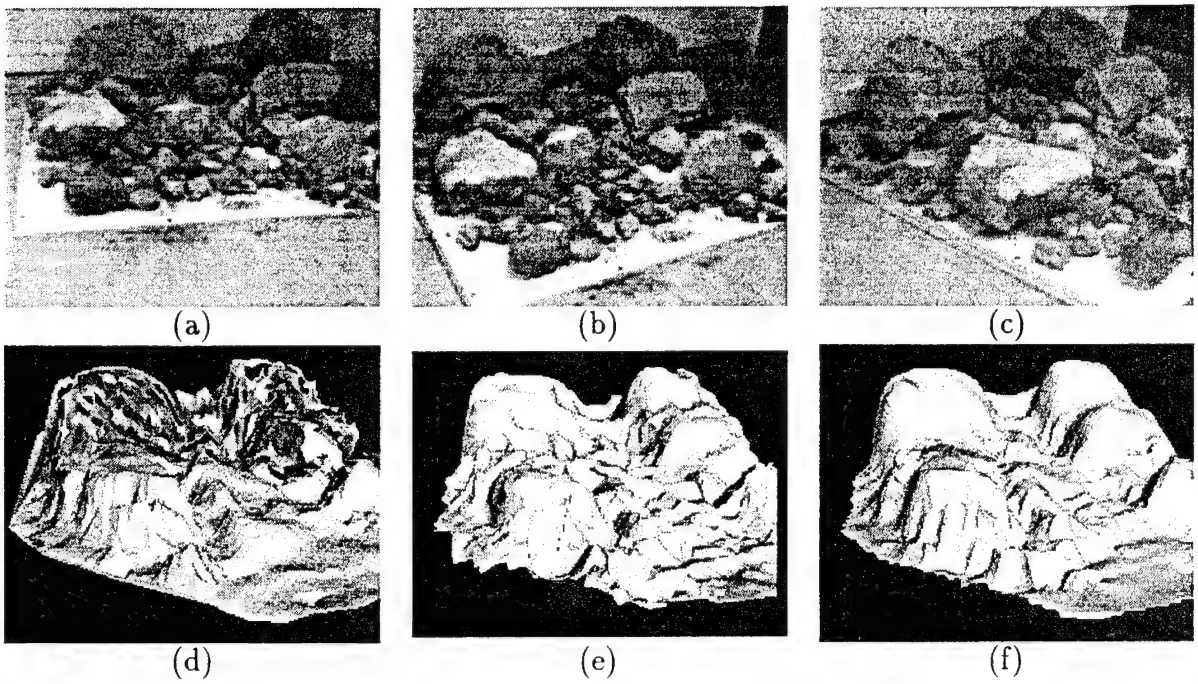


Figure 13: (a,b,c) The first images of three triplets taken by a mobile robot at different locations. (d) Rough ground level surface computed by combining correlation-based results from each triplet. (e) Optimized surface using stereo alone (f) Optimized surface using both stereo and shape from shading. (Courtesy of INRIA)

points, thereby allowing us to eliminate blindspots and make the reconstruction more robust where more than one view is available. The reconstruction process relies on both monocular shading cues and stereoscopic cues. We use these cues to drive an optimization procedure that takes advantage of their respective strengths while eliminating some of their weaknesses.

Specifically, stereo information is very robust in textured regions but potentially unreliable elsewhere. We therefore use it mainly in textured areas by weighting the stereo component most strongly for facets of the triangulation that project into textured image areas. The stereo component compares the gray-levels of the points in all of the images for which the projection of a given point on the surface is visible, as determined using a hidden-surface algorithm. This comparison is done for a uniform sampling of the surface. This method allows us to deal with arbitrarily slanted regions and to discount occluded areas of the surface.

On the other hand, shading information is mostly helpful in textureless areas. Thus, we weight the shading component most strongly for facets that project into textureless areas.

For utilizing shading information, the component uses a new method that does not invoke the traditional assumption of constant albedo. Instead, it attempts to minimize the variation in albedo across the surface, and can therefore deal with both constant albedo surfaces as well as surfaces whose albedo varies slowly. However, it does require the boundary conditions that are provided by the stereo information.

We have developed a weighting scheme that allows our system to use each source of information where it is most appropriate. As a result, for the large class of surfaces that roughly satisfy the Lambertian model, it performs significantly better than if it were using either source of information alone.

Here we have concentrated on Lambertian surfaces, where the image intensity of a surface point is independent of the direction from which it is viewed. Consequently, we have been able to directly use the intensity in both the stereo component of our algorithm, by using image-intensity correlation, and in the shape-from-shading component, by averaging the intensity at one surface point across all of the input images. Non-Lambertian surfaces cannot be dealt with in this manner.

In future work, we wish to extend our algorithm to the class of non-Lambertian surfaces for which it is possible to unambiguously compute the parameter(s) of the surface reflectance function given the image intensity, viewing direction, light-source direction, and surface normals (all of which are available during our optimization procedure, either directly or from the current estimate of the surface). An example of such a reflectance function is the model proposed by Oren and Nayar (1993) for known values of the surface roughness parameter. For this class of surfaces, we can use the estimated parameters to compute our objective function. For example, we can replace our intensity-correlation term by the variance of the estimated parameters across the input images at one surface point, and our shape-from-shading term by the average of these parameters across the input images at one surface point. This approach has the added advantage that it can be used in situations where the light-source direction is different for each image (as is often the case with aerial images).

Also in future work, we intend to investigate more complex topologies than the ones shown here, multiple resolutions and the shrinking or growing of the surface of interest. We have concentrated so far on a better understanding of the behavior of the objective function, but we believe that our approach can naturally support these extensions.

Acknowledgments

Support for this research was provided by various contracts from the Advanced Research Projects Agency. We wish to thank Hervé Matthieu and Olivier Monga, who have provided us with the face images and corresponding calibration data that appear in this paper and have proved extremely valuable to our research effort. We also thank the reviewers whose constructive criticisms have helped us improve the readability of the paper. Finally, we

would like to apologize to the members of the INRIA ROBOTVIS project, whose faces we have mercilessly deformed during the development of the algorithms discussed above.

References

Abbot, A. L. and Ahuja, N. (1990). Active surface reconstruction by integrating focus, vergence, stereo, and camera calibration. In *International Conference on Computer Vision*, pages 489–492.

Aloimonos, J. Y. (1989). Unification and integration of visual modules: an extension of the Marr paradigm. In *ARPA Image Understanding Workshop*, pages 507–551.

Asada, M., Kimura, M., Taniguchi, Y., and Shirai, Y. (1992). Dynamic integration of height maps into a 3D world representation from range image sequences. *International Journal of Computer Vision, 9(1)*, 31–54.

Baltsavias, E. P. (1991). *Multiphoto Geometrically Constrained Matching.* Ph.D. thesis, Institute for Geodesy and Photgrammetry, ETH Zurich.

Barnard, S. (1989). Stochastic stereo matching over scale. *International Journal of Computer Vision, 3(1)*, 17–32.

Barrow, H. G. and Tenenbaum, J. M. (1978). Recovering intrinsic scene characteristics from images. In *Computer Vision Systems*, pages 3–26, Academic Press, New York, New York.

Blake, A., Zisserman, A., and Knowles, G. (1985). Surface descriptions from stereo and shading. *Image Vision Computation, 3(4)*, 183–191.

Choe, Y. and Kashyap, R. L. (1991). 3-d shape from a shaded and textural surface image. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*, 907–919.

Cohen, I., Cohen, L. D., and Ayache, N. (1991). Introducing new deformable surfaces to segment 3D images. In *Conference on Computer Vision and Pattern Recognition*, pages 738–739.

Cryer, J. E., Tsai, P.-S., and Shah, M. (1992). *Combining shape from shading and stereo using human vision model.* Technical Report CS-TR-92-25, U. Central Florida.

Delingette, H., Hebert, M., and Ikeuchi, K. (1991). Shape representation and image segmentation using deformable surfaces. In *Conference on Computer Vision and Pattern Recognition*, pages 467–472.

Diehl, H. and Heipke, C. (1992). Surface reconstruction from data of digital line cameras by means of object based image matching. In *International Society for Photogrammetry and Remote Sensing*, pages 287–294, Washington D.C.

Faugeras, O. and Toscani, G. (1986). The calibration problem for stereo. In *Conference on Computer Vision and Pattern Recognition*, pages 15–20, Miami Beach, Florida.

Ferrie, F. P., Lagarde, J., and Whaite, P. (1992). Recovery of volumetric object descriptions from laser rangefinder images. In *European Conference on Computer Vision*, Genoa, Italy.

Fua, P. (1993). A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications, 6(1)*. Available as INRIA research report 1369.

Fua, P. and Leclerc, Y. G. (1990). Model driven edge detection. *Machine Vision and Applications, 3*, 45–56.

Fua, P. and Sander, P. (1992). Segmenting unstructured 3d points into surfaces. In *European Conference on Computer Vision*, Genoa, Italy.

Grimson, W. E. L. and Huttenlocher, D. P. (1992). Introduction to the special issue on interpretation of 3-d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2)*, 97–98.

Hannah, M. J. (1989). A system for digital stereo image matching. *Photogrammetric Engineering and Remote Sensing, 55(12)*, 1765–1770.

Hartt, K. and Carlotto, M. (1989). A method for shape-from-shading using multiple images acquired under different viewing and lighting conditions. In *Conference on Computer Vision and Pattern Recognition*, pages 53–60.

Heipke, C. (1992). Integration of digital image matching and multi image shape from shading. In *International Society for Photogrammetry and Remote Sensing*, pages 832–841, Washington D.C.

Horn, B. K. P. (1990). Height and gradient from shading. *International Journal of Computer Vision, 5(1)*, 37–75.

Hung, Y., Cooper, D. B., and Cernuschi-Frias, B. (1991). Asymptotic bayesian surface estimation using an image sequence. *International Journal of Computer Vision, 6(2)*, 105–132.

Kaiser, B., Schmolla, M., and Wrobel, B. P. (1992). Application of image pyramid for surface reconstruction with fast vision. In *International Society for Photogrammetry and Remote Sensing*, page 1, Washington, D.C.

Kanade, T. and Okutomi, M. (1990). A stereo matching algorithm with an adaptative window: Theory and experiment. In *ARPA Image Understanding Workshop.*

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision, 1(4)*, 321–331.

Leclerc, Y. G. (1989a). Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision, 3(1)*, 73–102.

Leclerc, Y. G. (1989b). *The Local Structure of Image Intensity Discontinuities.* Ph.D. thesis, McGill University, Montréal, Québec, Canada.

Leclerc, Y. G. and Bobick, A. F. (1991). The direct computation of height from shading. In *Conference on Computer Vision and Pattern Recognition*, Lahaina, Maui, Hawaii.

Liedtke, C. E., Busch, H., and Koch, R. (1991). Shape adaptation for modelling of 3D objects in natural scenes. In *Conference on Computer Vision and Pattern Recognition*, pages 704–705.

Lowe, D. G. (1991). Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(441-450).*

Luenberger, D. G. (1984). *Linear and Nonlinear Programming.* Addison-Wesley, Menlo Park, California, second edition.

Marr, D. (1982). *Vision.* W. H. Freeman, San Francisco, California.

Nishihara, H. (1984). Practical real-time imaging stereo matcher. *Optical Engineering, 23(5).*

Okutomi, M. and Kanade, T. (1991). A multiple-baseline stereo. In *Computer Vision and Pattern Recognition 91*, pages 63–69, Maui, Hawaii.

Oren, M. and Nayar, S. (1993). Generalization of the lambertian model. In *ARPA Image Understanding Workshop*, pages 1037–1048.

Panton, D. J. (1978). A flexible approach to digital stereo mapping. *Photogramm. Eng. Remote Sensing, 44(12)*, 1499–1512.

Pentland, A. (1990). Automatic extraction of deformable part models. *International Journal of Computer Vision, 4(2)*, 107–126.

Pentland, A. and Sclaroff, S. (1991). Closed-form solutions for physically based shape modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13*, 715–729.

Poggio, T., Torre, V., and Koch, C. (1985). Computational vision and regularization theory. *Nature, 317*.

Press, W., Flannery, B., Teukolsky, S., and Vetterling, W. (1986). *Numerical Recipes, the Art of Scientific Computing.* Cambridge U. Press, Cambridge, MA.

Quam, L. (1984). Hierarchical warp stereo. In *ARPA Image Understanding Workshop*, pages 149–155.

Stokely, E. M. and Wu, S. Y. (1992). Surface parameterization and curvature measurement of arbitrary 3-d objects: five practical methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(8)*, 833–839.

Szeliski, R. (1991). Shape from rotation. In *Conference on Computer Vision and Pattern Recognition*, pages 625–630.

Szeliski, R. and Tonnesen, D. (1992). Surface modeling with oriented particle systems. In *Computer Graphics (SIGGRAPH'92)*, pages 185–194.

Terzopoulos, D. (1986). Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 8*, 413–424.

Terzopoulos, D. (1988). The computation of visible-surface representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, , 417–438.

Terzopoulos, D. and Metaxas, D. (1991). Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(703-714)*.

Terzopoulos, D. and Vasilescu, M. (1991). Sampling and reconstruction with adaptive meshes. In *Conference on Computer Vision and Pattern Recognition*, pages 70–75.

Terzopoulos, D., Witkin, A., and Kass, M. (1987). Symmetry-seeking models and 3D object reconstruction. *International Journal of Computer Vision, 1*, 211–221.

Tomasi, C. and Kanade, T. (1992). The factorization method for the recovery of shape and motion from image streams. In *ARPA Image Understanding Workshop*, pages 459–472.

Vemuri, B. C. and Malladi, R. (1991). Deformable models: Canonical parameters for surface representation and multiple view integration. In *Conference on Computer Vision and Pattern Recognition*, pages 724–725.

Wang, Y. F. and Wang, J. F. (1992). Surface reconstruction using deformable models with interior and boundary constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(5)*, 572–579.

Whaite, P. and Ferrie, F. P. (1991). From uncertainty to visual exploration. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(1038-1049).*

Witkin, A. W., Terzopoulos, D., and Kass, M. (1987). Signal matching through scale space. *International Journal of Computer Vision, 1*, 133–144.

Wrobel, B. P. (1991). The evolution of digital photgrammetry from analytical phogrammetry. *Photogrammetric Record, 13(77)*, 765–776.

Appendix D

"Reconstructing Complex Surfaces from Multiple Stereo Views"

ICCV, Boston, MA, June 1995

Pascal V. Fua

Reconstructing Complex Surfaces from Multiple Stereo Views

P. Fua
SRI International
333 Ravenswood Avenue, Menlo Park, CA 94025, USA
(fua@ai.sri.com)

Abstract

We present a framework for 3–D surface reconstruction that can be used to model fully 3–D scenes from an arbitrary number of stereo views taken from vastly different viewpoints. This is a key step toward producing 3–D world-descriptions of complex scenes using stereo and is a very challenging problem: real-world scenes tend to contain many 3–D objects, they do not usually conform to the 2-1/2–D assumption made by traditional algorithms, and one cannot take it for granted that the computed 3–D points can easily be clustered into separate groups. Furthermore, stereo data is often incomplete and sometimes erroneous, which makes the problem even more difficult.

By combining a particle-based representation, robust fitting, and optimization of an image-based objective function, we have been able to reconstruct surfaces without any *a priori* knowledge of their topology and in spite of the noisiness of the stereo data.

Our current implementation goes through three steps—initializing a set of particles from the input 3–D data, optimizing their location, and finally grouping them into global surfaces. Using several complex scenes containing multiple objects, we demonstrate its competence and ability to merge information and thus to go beyond what can be done with conventional stereo alone.

1 Introduction

Reconstructing arbitrary 3–D surfaces from stereo image pairs—or more generally N-tuplets—remains an unsolved problem because they provide incomplete, and sometimes erroneous, information about the location of 3–D points in space. Grouping these points into meaningful surfaces involves solving a problem akin to the notoriously hard segmentation problem. As a result, a majority of recent computer vision approaches to surface reconstruction rely on much cleaner sources of 3–D data—such as laser range maps or medical volumetric data—as their input. Many of these approaches also assume that there is one, and only one, object of interest whose topology is known in advance so that a particular model—be it rigid or deformable—can be fit to the data.

Stereo, however, is a purely passive technique that only involves the use of relatively inexpensive and dependable sensors and often is the only feasible approach. We propose a novel alternative to conventional stereo reconstruction that overcomes many of the difficulties that traditional approaches encounter in the reconstruction of 3–D surfaces from stereo imagery.

More specifically, to recover surfaces from stereo, one must contend with the following problems:

- Real-world scenes often contain several objects whose topology may not be known in advance: some surfaces are best modeled as sheets while others are topological spheres or contain holes. One cannot typically assume that there is only one object and one surface of interest or expect to be able to easily cluster the 3–D points derived from stereo into semantically meaningful groups.
- The 2-1/2–D assumption that most traditional interpolation schemes make is no longer valid when reconstructing complex 3–D scenes from arbitrary numbers of images and arbitrary viewpoints. Surfaces often overlap and may be visible in one view but not another.
- The 3–D points derived from disparity maps form an irregular sampling of the measured surfaces. In addition, small errors in disparity can result in large errors in world position.
- Even the best stereo algorithms make occasional blunders that must be identified and eliminated. Furthermore, the corresponding erroneous 3–D points are often correlated with one another so that they cannot be eliminated by robust estimation alone.

To address these problems, our approach

- Relies on an object-centered representation that can handle surfaces of arbitrary complexity, specifically a set of connected surface patches or "oriented particles" as defined by Szeliski and Tonnesen [1992].
- Refines the surface's description by minimizing an objective function that combines terms measuring both surface smoothness and gray scale correlation in the input images of the surface points'projections.

We typically start with a set of stereo pairs or triplets. We compute disparity maps for each of them, fit local quadric surface patches to the corresponding 3–D points, and use these patches to instantiate a set of particles. We then refine their positions by minimizing the objective function, thereby returning to the original image data. Finally, we impose a metric upon the set of particles that allows us to cluster them into meaningful global surfaces.

Our technique allows the modeling of complex 3–D scenes whose topology is unknown *a priori* from stereo data without the need to rely on other sources of range data. This ability is valuable for applications, such as robotics and high-resolution cartography, where precise scene models are not always available. For example a mobile robot must be able to distinguish the ground from objects lying on it to model obstacles and avoid them. Similarly, to grasp and manipulate objects, a robot must be able to distinguish and model them. In the case of low-resolution cartography, the earth's surface can indeed be relatively well modeled as a single surface. At high-resolution, however, this stops being true: objects such as trees or buildings must be modeled as separate 3–D surfaces, and the usual 2-1/2–D assumptions break down.

In the following section, we describe related work and our contributions in this area. We then present our framework in detail, discuss the procedure's behavior on synthetic data, and show results on real images of complex scenes.

2 Related Work and Contributions

Many recent publications describe the reconstruction of a surface using 3–D object-centered representations, such as 2-1/2–D grids [Robert *et al.*, 1992], 3–D surface meshes [Cohen *et al.*, 1991, Delingette *et al.*, 1991, Terzopoulos and Vasilescu, 1991, Vemuri and Malladi, 1991, McInerney and Terzopoulos, 1993, Koh *et al.*, 1994, Chen and Medioni, 1994], parameterized surfaces [Stokely and Wu, 1992, Lowe, 1991], local surfaces [Sander and Zucker, 1990, Ferrie *et al.*, 1992, Stewart, 1994], particle systems [Szeliski and Tonnesen, 1992], spanning trees [Hoppe *et al.*, 1992], and volumetric models [Pentland, 1990, Terzopoulos and Metaxas, 1991, Pentland and Sclaroff, 1991, Park *et al.*, 1994]. Most of these rely on previously computed 3–D data, such as the coordinates of points derived from laser range finders or correlation-based stereo algorithms, and reconstruct the surface by fitting it to these data in a least-squares sense. In other words, the derivation of the 3–D data is completely divorced from the reconstruction of the surface. Errors and imprecisions in the original 3–D input data can jeopardize irretrievably the quality of the reconstruction.

In previous work, we began addressing this issue by developing an approach to 3–D surface reconstruction that uses image cues while recovering the surface's shape [Fua and Leclerc, 1994a, Fua and Leclerc, 1994b]. It uses an object-centered representation—specifically a 3–D triangulated mesh—and can take advantage of both monocular shading cues and stereoscopic cues from any number of images to refine the model while correctly handling self-occlusions.

However, in our previous approach—as in most of those mentioned above—it is assumed that the range data used to initialize the models can easily be clustered into separate groups that define distinct objects. A separate 3–D model can then be fitted to each object. For complex scenes, this clearly is much too strong an assumption. The technique presented in this paper, was specifically designed to eliminate the need for this assumption. It replaces our triangulated meshes by a particle system that does not require any *a priori* knowledge of the surface's topology and lends itself to the definition of a metric that has allowed us to effectively cluster local surface patches into global ones.

The particles can adjust so as to minimize an objective function that is the sum of an image-based term and of a regularization term. The image-based term is a generalization of the multi-image intensity correlation term we used in our mesh work, except for the fact that we replace the triangular facets by circular surface patches attached to each particle. Optimization allows us to refine the position of legitimate particles and to eliminate spurious ones. Consequently, we do not have to depend on the input 3–D data to be error-free to achieve good results. This is in contrast to Szeliski and Tonnesen's particles [1992] that have been used to great effect to model high quality medical data but make no provisions for noisy and incorrect data points.

To instantiate our set of particles, we robustly fit local surfaces to the raw disparity data in a manner that is closely related to other local surface techniques e.g., [Sander and Zucker, 1990, Ferrie *et al.*, 1992, Hotz *et al.*, 1993, Stewart, 1994]. In our approach, however, these local surfaces are not the end result since they can be refined and either accepted as part of a global surface or rejected. In spirit, our approach is closely related to that advocated by Hoff and Ahuja [1987] because it combines the processes of feature matching and surface interpolation. Unlike this earlier work, however, our technique is not limited to pairs of images.

We view the central contribution of this paper as twofold. First, we provide a specific alternative framework that explicitly addresses many of the problems of conventional stereo and a representation in which they can be solved. Second, we demonstrate that our implementation of this framework actually solves some of the problems involved in reconstructing complex 3–D surfaces of unknown topology and provides a foundation on which to build a solution for the others.

3 From Raw Stereo Data to Global Surfaces

Our approach to recovering complex surfaces is to model them as sets of local surface elements that interact with one another. Following Szeliski and Tonnesen [1992], we refer to the surface patches as "oriented particles." The forces that bind them can be understood as "breakable springs" [Terzopoulos and Vasilescu, 1991] that tend to align the particles with each other but may break for particles that are too far out of alignment.

We use several sets of stereo pairs or triplets of a given scene as our input data. We assume that the images are monochrome and registered so that their relative camera models are known *a priori*. Since we are interested in

reconstructing surfaces, we start the process by using a simple correlation-based algorithm [Fua, 1993] to compute a disparity map for each pair or triplet and by turning each valid disparity value into a 3–D point. If other sources of range data were available, they could be used in a similar fashion. These 3–D points typically form an extremely noisy and irregular sampling of the underlying global 3–D surfaces. To effectively model these, we take the following steps:

1. Robustly fit particles to the raw 3–D points.
2. Refine the position and orientation of the particles by minimizing an image-based objective function.
3. Eliminate spurious particles and cluster those that appear to belong to the same global surfaces.

In the remainder of this section we first introduce our particles and then describe our technique in more detail.

3.1 Oriented Particles

Our surface elements are disks whose geometry is defined by the positions of their centers, the orientations of their normals and their radii. In theory, these disks have six degrees of freedom. However, when modeling a global surface in terms of such disks, translations along the tangent plane of the surface can be ignored as long as the disks remain roughly equidistant from one another and the radius can be chosen so that the disks approximately cover the surface. Therefore, in practice, we deal with only three degrees of freedom.

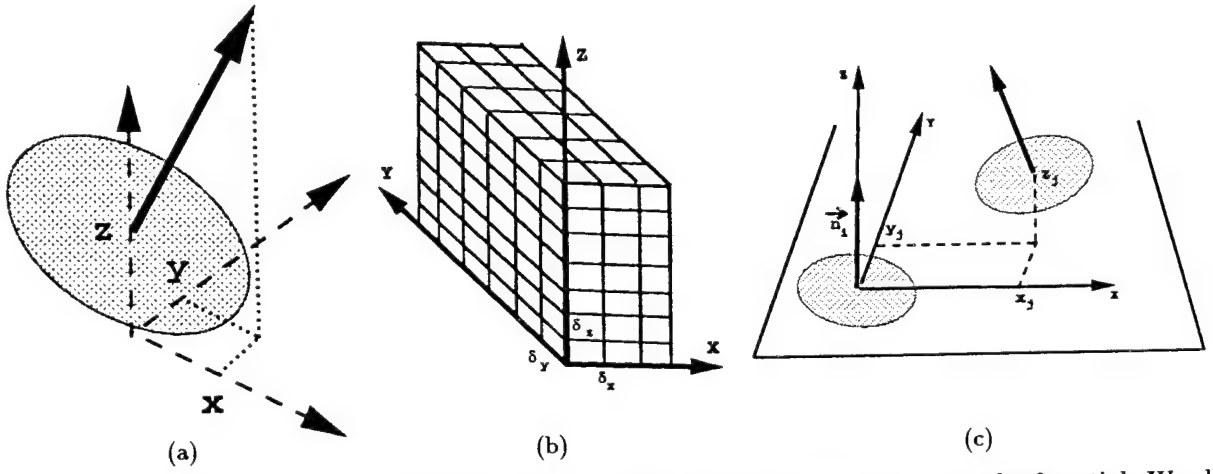


Figure 1: Data structures and metric. (a) A particle is a disk to which we associate a local referential. We allow the center of gravity to shift along the z axis and parametrize the orientation using the projections of the normal vector on the x and y axes. (b) The input 3–D points are stored in a cube-shaped set of voxels and we instantiate a particle in each voxel containing enough such points. (c) The "distance" between two particles is primarily a function of the distance of the center of gravity of one particle from the tangent plane of another.

As shown in Figure 1, to achieve an orientation-independent implementation, we assign to each particle a local referential. As discussed in Section 3.2, we instantiate this referential by robustly fitting surface patches to the 3–D points within local neighborhoods and using the surface normals to define the local z directions. We define a particle's position by the translation of the center along the local vertical and its orientation by the x and y projections of the normal on the local x and y axes. This particular parametrization is most favorable when the local vertical is relatively close to the normal of the surface under consideration for two reasons. First, the x and y projections of the particle's normal vector will then be relatively small and the interaction forces between particles almost quadratic in terms of those parameters. Second, displacements along the local z axis will be close to being normal to the underlying surface and thus precisely the ones that are most significant in terms of recovering its shape.

3.2 Initialization

To generate a set of regularly spaced particles from our noisy stereo data, we pick spatial step sizes δ_x,δ_y and δ_z along the X,Y and Z axes of an absolute referential. We use them to define a cube-shaped set of 3–D buckets, such as the one of Figure 1(b). We then store the 3–D points computed from our initial correlation data into the appropriate buckets. By fitting a local surface to every bucket containing enough points, we generate particles whose center is the projection of the bucket's center onto the surface and whose orientation is given by the surface's normal at that point. In the presence of very noisy data, the projection may fall outside of the bucket. In this case, we reject the particle thereby, ensuring that there is only one particle per bucket and that the particles are regularly distributed.

In general, most of the 3–D buckets will be empty. Therefore we do not store the set of 3–D buckets as a cube but as a hash-table allowing for both compact storage and easy computation of neighborhood relationships.

For the initialization phase to be successful, it is important both to choose the right kind of surface model and to use a robust method to perform the fitting. We have used both planar and quadric models. The quadrics, even though they involve more computation, have proven very effective because they allow the use of larger sets of points than planes without introducing any appreciable bias. In our implementation, we take advantage of this by fitting, to each bucket containing enough points, a plane of form

$$ax + by + cz = h$$

when x,y and z are coordinates in the absolute referential. We then fit a quadric of form

$$z' = ax'x' + bx'y' + cy'y' + dx' + ez' + f$$

where x',y' and z' are coordinates defined by the plane. We use not only the points in the bucket under consideration but also in the buckets that are its immediate neighbors. This method allows us to fit local surfaces of arbitrary orientation using a relatively large set of 3–D points, and tends to enforce consistency of orientation among neighboring particles.

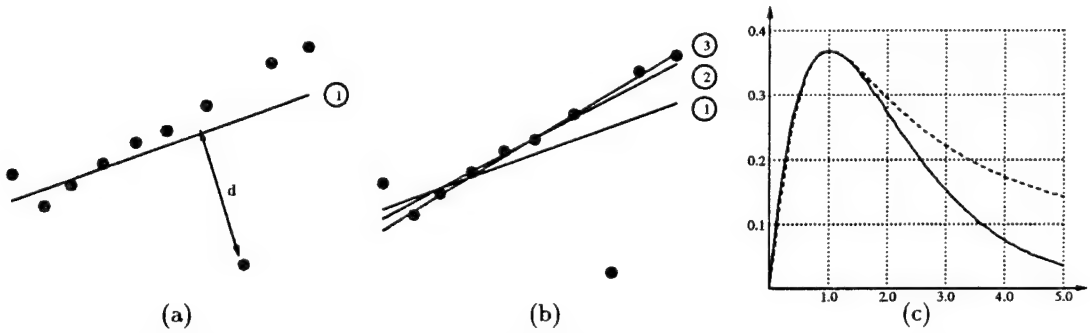


Figure 2: Iterative Reweighted Least Squares. (a) We first use standard least squares to fit an initial surface patch to the set of noisy points and compute the distance d of these data points to the fit. The points are denoted by the black circles, and the initial patch is labeled 1. (b) We weigh each data point inversely proportionally to d, fit a new surface, and iterate the procedure, thereby generating the fits labeled 2 and 3. After a few iterations, the contribution of the outliers becomes negligible. (c) The influence function of the data points is shown as a solid line. For comparison's sake, a Lorentzian influence function, scaled so as to have the same maximum, is shown as a dashed line.

Because of the noisiness of the input data, a robust surface-fitting method is essential. In this implementation, we use a variant of the Iterative Reweighted Least Squares [Beaton and Turkey, 1974] technique, as illustrated by Figure 2. To fit a surface of equation $f(x, y, z) = 0$ to a set $(x_i, y_i, z_i)_{1 \leq i \leq n}$ of n 3–D points, we minimize a criterion of the form

$$\sum_{1 \leq i \leq n} w_i f(x_i, y_i, z_i)^2 \qquad (1)$$

where the w_i are weights. At the first iteration the w_i are all taken to be equal to one so that the first fit, f_0, is a regular least-squares fit. We use f_0 to compute a new set of weights

$$\begin{aligned} w_i &= exp(\frac{-|f_0(x_i, y_i, z_i)|}{\overline{d}}) \quad \text{for } 1 \leq i \leq n \\ \overline{d} &= \text{median}|f_0(x_i, y_i, z_i)|_{1 \leq i \leq n} \end{aligned} \tag{2}$$

and to fit a new surface f_1 by reminimizing the criterion of Equation 1. In effect, we use the median distance of the points to the fitted surface as an estimate of the noise variance, and we discount the influence of points that are more than a few standard deviations away. This approach can be related to the use of Lorentzian estimators [Black and Rangarajan, 1994] because the influence function of the data points, shown in Figure 2(c), has roughly the same shape. However, it requires no *a priori* estimate of the variance of the noise and involves only least-squares, and therefore fast, minimization.

By iterating this procedure a few—typically five—times, we have been consistently able to fit our quadric surfaces to noisy data, even in the presence of large numbers of outliers. Figure 3 illustrates the robustness of our algorithm.

3.3 Clustering

To cluster the isolated particles into more global entities, we define a simple "same surface" relationship $\mathcal{R}$ between particles P_i and P_j as follows:

$$P_i \mathcal{R} P_j \iff d_{\text{part}}(P_i, P_j) < \max_d \tag{3}$$

where d_{part} is a distance function and $\max_d$ a distance threshold. We could take d_{part} to be the Euclidean distance between particle centers. However, such a distance would not be discriminating enough for our purposes because it is circularly symmetric and does not take the particles'orientation into account. It has proved much more effective to define a distance function that penalizes more heavily the distance of one particle's center from the tangent plane of the other than the distance along the tangent plane. The simplest way to achieve this result is to define d_{part} as follows:

$$\begin{aligned} d_j &= k z_j^2 + (1-k)(x_j^2 + y_j^2) \\ d_i &= k z_i^2 + (1-k)(x_i^2 + y_i^2) \\ d_{\text{part}}(P_i, P_j) &= max(d_i, d_j) \end{aligned} \tag{4}$$

where x_j,y_j and z_j are the coordinates of the center of P_j in a referential whose Z direction is the normal of P_i and whose origin is the center of P_i, as shown in Figure 1(c), and k is a constant larger than 0.5. In this paper, we take $k = 0.9$; x_i,y_i and z_i are defined symmetrically.

In essence, the threshold $\max_d$ on d_{part} limits the curvature of the common underlying surface to which particles may belong. As such it is domain dependent; here we take $\max_d$ to be a multiple, typically 1.5, of the median value of d_{part} for all pairs of neighboring particles in the cube-shaped structure of Section 3.2.

The data set equipped with the relationship $\mathcal{R}$ can now be viewed as a graph whose connected components are the surfaces we are looking for. In practice, there will usually be spurious particles that are weakly linked to legitimate clusters. In such cases, we have found that removing all points that do not have a minimum number of neighbors allows us to throw away the gross errors and generate meaningful clusters such as the ones depicted in the last column of Figure 3.

3.4 Refining

Because it is extremely difficult to design a stereo algorithm that never produces correlated artifacts, we cannot expect any robust fitting technique to exclude all erroneous 3–D points. Furthermore, fitting local surfaces to the initial data amounts to smoothing and may result in spurious particles that appear to line up with legitimate ones and become very hard to eliminate.

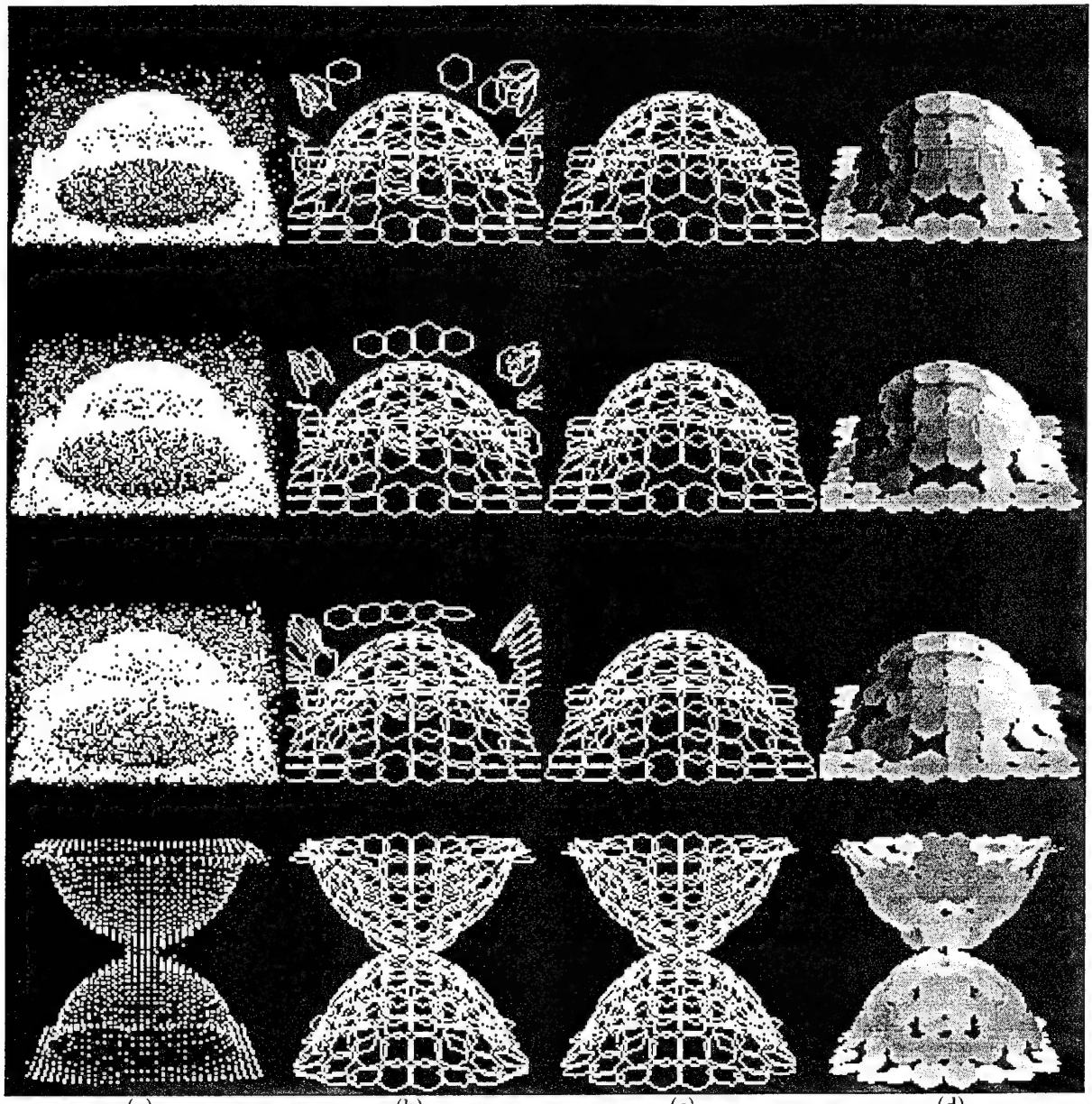


Figure 3: Initializing particle sets. (a) Four sets of noisy 3–D points. In the first three, the majority of points—90%, 80% and 60%, respectively—lie on a hemisphere while the remaining ones—10%, 20% and 40%, respectively—are uniformly randomly distributed. In the fourth set, the points lie on two hemispheres. (b) The particles instantiated by our robust fitting procedure. Note that there is not a particle in every voxel because, as explained in the text, the obviously meaningless ones are eliminated. (c) The particles that remain after clustering. All surviving spurious particles have disappeared. (d) A shaded view of the same particles. They are rendered as lambertian planar patches.

To illustrate this point and to produce a difficult example that gives rise to some of the same problems we have encountered when dealing with real imagery, we have generated the five synthetic images of a textured hemisphere occluding a plane shown in Figure 4. Our intent was to produce images ambiguous enough for a conventional correlation algorithm to compute spurious disparities and to show that our refinement procedure provides the additional power required to eliminate them.

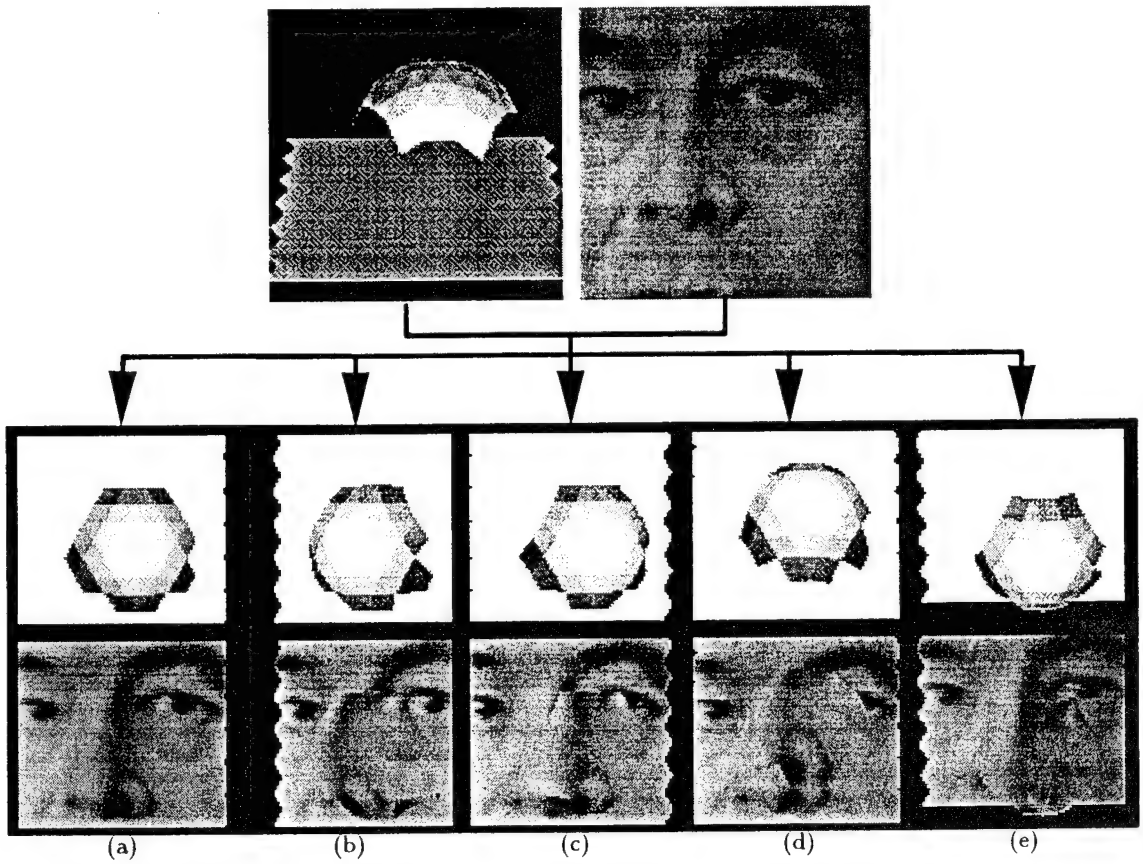


Figure 4: Constructing synthetic images of a hemisphere occluding a plane. We begin with the triangulated mesh and real face image shown in the top row. We then consider the five views depicted by the middle row and texture map the real image onto the triangulated mesh to produce the five synthetic images of the bottom row. Obviously, these five images would have been easier to parse had we mapped a different texture on the plane and the hemisphere. However, our intention in generating these images was to produce a difficult example that gives rise to some of the same problems we have encountered when dealing with real imagery.

On the left of the first row of Figure 5, we show a noise-free set of 3–D points that belong to either the plane or the hemisphere and, on the right side, the particles that are instantiated by running our initialization procedure on this set. In the leftmost column of the following rows, we show three subsets of all the 3–D points generated by correlating pairs of these images—specifically (a) and (b), (a) and (c), (a) and (d), (a) and (e)—using three different window sizes, 5x5 for the first row, 10x10 for the second, and 15x15 for the third. The second column from the left depicts the particles that are instantiated using this data.

As the window size increases, the 3–D points appear to be smoother but, in fact, fit the data less well. Of course this problem can be alleviated by using variable-shaped windows [Nishihara, 1984, Devernay and Faugeras, 1994]

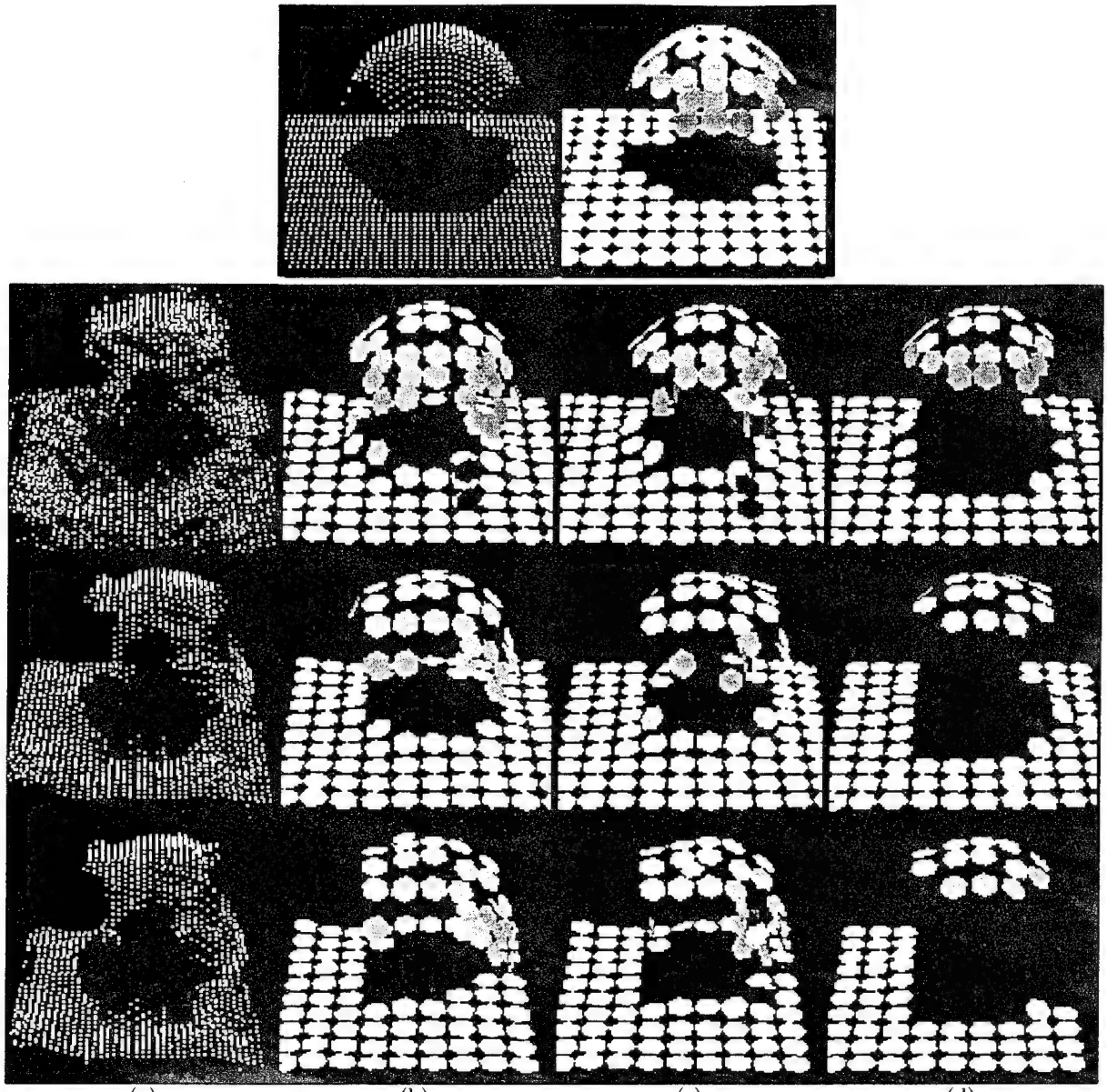


Figure 5: Running our complete procedure on the images of Figure 4. Top row: A set of 3–D points lying on the surfaces used to generate the images and a shaded view of the particles that our procedure fits to these points. Column (a): 3–D points computed by running a correlation-based algorithm on the synthetic images by using, from top to bottom, fixed windows of size 5x5, 10x10, and 15x15. Column (b): The particles instantiated from these sets of points. Column (c): The same particles after optimization. Column (d): For each set of particles, the two subsets that appear to belong to the same surface according to our metric.

and, our approach to particle refinement can be viewed as a 3–D generalization of these purely image-based techniques.

A more serious problem stems from the presence of correlated but erroneous 3–D points on the right side of the shape. These points are produced by the correlation windows that straddle the hemisphere and the plane, and line up well enough so that the fitting procedure produces a set of surface patches that appear to be valid but do not correspond to any physical surface.

To resolve such problems, it is necessary to return to the original images and assess the quality of each particle. For each disk-shaped particle, we define a "multi-image intensity correlation" term by projecting the 3–D disks into 2–D elliptical patches in each image and measuring how well these patches correlate. This term is similar to the one we defined in previous work for 3–D triangular facets [Fua and Leclerc, 1994a]. It is a function of the three degrees of freedom of each particle and can therefore be used to perform optimization.

When using noise free synthetic images such as the ones of Figure 4, we have verified experimentally that, for legitimate particles, the minimum of this image-correlation term occurs for positions and orientations that are very close to the ones computed by simply fitting the disparity maps while the position of spurious particles may be arbitrarily far from the minimum. In such an ideal case, the good particles could be separated from the bad ones on that basis alone. However, when dealing with real images that are never entirely noise free, the situation becomes more complex and the minima for the individual particles may shift substantially because of the noise. A more practical approach is then to allow the particles to interact with one another and to rearrange themselves to minimize an energy term that is the sum of the multi-image intensity correlation term discussed above and of a deformation energy term that tends to enforce consistency between neighboring particles [Szeliski and Tonnesen, 1992]. As illustrated by the two rightmost columns of Figure 5, while optimizing the energy term, the particles that actually correspond to the same underlying global surfaces will "stick together" and the ones that do not will tend to move in separate directions, stop lining up with each other and be easily eliminated by the clustering technique of Section 3.3.

Formally, we write the total energy of a set of particles $\mathcal{E}_T$ as

$$\mathcal{E}_T = \mathcal{E}_{St} + \lambda_D \mathcal{E}_D \ , \tag{5}$$

where $\mathcal{E}_{St}$ is the multi-image intensity correlation term, $\mathcal{E}_D$ the deformation energy term, and λ_D a weighting coefficient.

We now turn to the formal definition of these energy terms.

3.4.1 Multi-Image Intensity Correlation

The basic premise of most correlation-based stereo algorithms is that the projection of the 3–D points into various images, or at least band-passed or normalized versions of these images, must have identical gray levels. To take advantage of this property in our particle-based representation, we define the stereo component of our objective function as the variance in gray-level intensity of the projections in the various images of a given sample point on a particle, summed over all sample points, and summed over all particles. This component is depicted by Figure 6(a) and is presented in stages below.

First, we define the sample points of a disk-shaped particle P, of center $\mathbf{x}_0$, radius R and normal $\vec{n}$, by noting that all points on a particle can be written as

$$\mathbf{x}_{r,\theta} = \mathbf{x}_0 + r\cos(\theta)\vec{v_x} + r\sin(\theta)\vec{v_y} \ \text{ for } 0 \leq r \leq R \text{ and } 0 \leq \theta \leq 2\pi \ , \tag{6}$$

where $\vec{v_x}$ and $\vec{v_y}$ are unit vectors chosen so that $\vec{v_x}$,$\vec{v_y}$ and $\vec{n}$ form an orthonormal basis. We obtain our regularly spaced sample points $\mathbf{x}_{r_i,\theta_j}$ by taking

$$\begin{aligned} r_i &= i\frac{R}{n_r} \ \ 1 \leq i \leq n_r \\ \theta_j &= j\frac{2\pi}{n_\theta(r_i)} \ \ 1 \leq j \leq n_\theta(r_i) \end{aligned}$$

where $n_\theta(r_i) = 2\pi\frac{r_i}{R}n_r$ so as to ensure approximately uniform sampling of the disk.

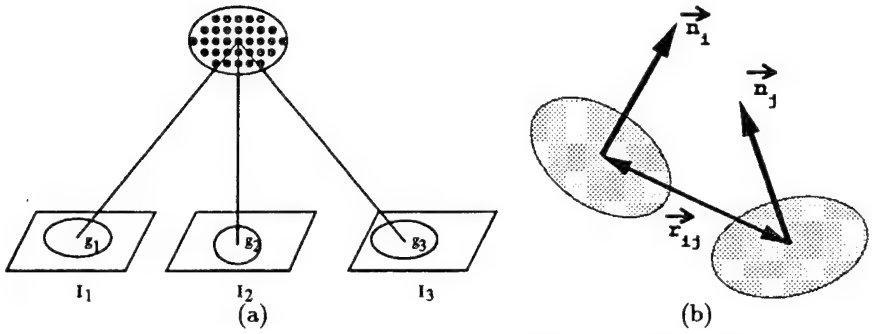


Figure 6: Computing the energy terms. (a) Particles are sampled uniformly; the circles represent the sample points. The stereo component of the objective function is computed by summing the variance of the gray level of the projections of these sample points, the g_is. (b) The forces that bind the particles are computed as functions of their normal vectors, $\vec{v_i}$ and $\vec{v_j}$, and of the vector $\vec{r_{ij}}$ whose endpoints are the centers of gravity.

Next, we develop the sum of squared differences in intensity from all images for a given point $\mathbf{x}$. A point $\mathbf{x}$ in space is projected into a point $\mathbf{u}$ in image g_i via the perspective transformation $\mathbf{u} = \mathbf{m}_i(\mathbf{x})$. Consequently, the sum of squared differences in intensity from all the images, $\sigma^2(\mathbf{x})$, is defined by

$$\mu(\mathbf{x}) = \frac{1}{n_i}\sum_{i=1}^{n_i} g_i(\mathbf{m}_i(\mathbf{x})) \ ,$$

$$\sigma^2(\mathbf{x}) = \frac{1}{n_i}\sum_{i=1}^{n_i} \left(g_i(\mathbf{m}_i(\mathbf{x})) - \mu(\mathbf{x})\right)^2 \ .$$

Note that $\mu(\mathbf{x})$ can be considered as the "gray level" of the sample point and by taking its median value over all sample points of the particle, we can define the median gray level of a particle that we use for clustering purposes as discussed in Section 3.5. Note also, that when using only two images, $\sigma^2(\mathbf{x})$ reduces to the square difference in gray levels. In this case, our approach is equivalent to straightforward correlation with deformable windows but has the advantage of generalizing to arbitrary numbers of images.

The above definition of $\sigma^2(\mathbf{x})$ does not take into account occlusions of the surface: not all sample points of all particles are visible in all images. In theory, this could be handled by generalizing the Z-buffering technique we used in previous work [Fua and Leclerc, 1994a]. However, in the current implementation we use a simpler heuristic. As discussed in Section 3.2, the particles are initialized by computing disparity maps using pairs or triplets of images and fitting local surfaces to the corresponding 3–D points. We record the origin of these points and associate to each particle the set of two or three images that provided the largest amount of support for the local surface. We use these images to compute $\sigma^2(\mathbf{x})$ because they are the ones in which the particle is most likely to be entirely visible.

The multi-image intensity correlation component $\mathcal{E}_{St}$ then becomes:

$$\mathcal{E}_{St} = \sum_k \mathcal{E}_{St}^k \tag{7}$$

$$\mathcal{E}_{St}^k = \frac{\sum_{r,\theta} \sigma^2(\mathbf{x}_{r,\theta})}{s_k} \ , \tag{8}$$

where s_k is the total number of samples for particle k, E_{St}^k is the value of the correlation component for a single particle and the summation over k denotes a summation over the set of all particles.

3.4.2 Particle Interactions

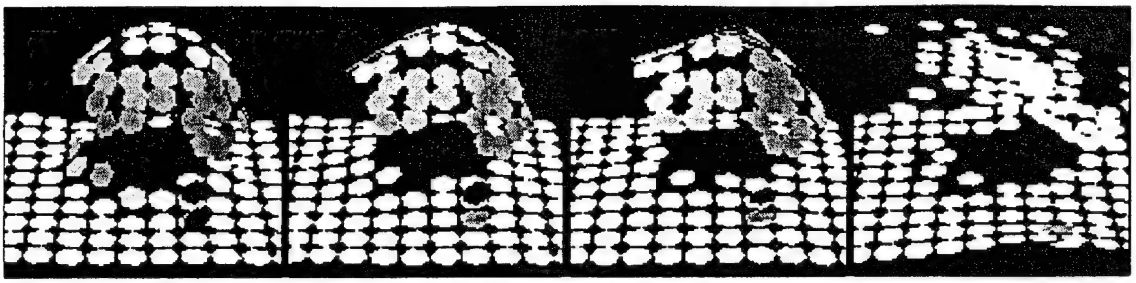


Figure 7: Rearranging the particles of the second row of Figure 5 by minimizing the deformation energy alone. The optimization progresses from left to right. All particles, but only a few outliers, end up lying on one of two separate planes.

Following Szeliski and Tonnesen [1992], we define a conormality potential $\mathcal{E}_{cn}^{ij}$ and a coplanarity potential $\mathcal{E}_{cp}^{ij}$ between particles i and j by writing

$$\begin{aligned} \mathcal{E}_{cn}^{ij} &= 1/2||\vec{n_i} - \vec{n_j}||^2 = 1 - \vec{n_i}\vec{n_j} \ , \\ \mathcal{E}_{cp}^{ij} &= 1/2((\vec{n_i}\vec{r_{ij}})^2 + (\vec{n_j}\vec{r_{ij}})^2) \ , \end{aligned} \tag{9}$$

where $\vec{n_i}$ and $\vec{n_j}$ are the normal vectors and $\vec{r_{ij}}$ the vector joining the centers of the two particles, as shown in Figure 6(b). These terms control the surface's resistance to bending and we take our overall regularization terms $\mathcal{E}_D$ to be

$$\mathcal{E}_D = \sum_{i,j} f(\mathcal{E}_{cn}^{ij} + \mathcal{E}_{cp}^{ij}) \tag{10}$$

where the summation over i and j denotes a summation over all pairs of particles that are neighbors in the cube-shaped structure of Section 3.1, and f is a monotonically increasing function. In practice we implement the concept of breakable springs by taking f to be

$$f(x) = log(1 + x/s)$$

with s being a fixed constant so that, as in Section 3.2, the interaction forces have a Lorentzian behavior [Black and Rangarajan, 1994]. As the particles move out of alignment, the strength of the interaction increases up to a point after which the interaction strength decreases and eventually vanishes.

In our implementation, we have not found it necessary to weight the interactions: by construction, our particles tend to be equidistant and cannot slide along the surfaces because they have only three degrees of freedom. In Figure 7, we show the result of rearranging one of the sets of particles from Figure 5 by minimizing $\mathcal{E}_D$ alone.

3.5 Global Optimization

Recall that the total energy of a set of n particles is written as

$$\mathcal{E}_T = \mathcal{E}_{St} + \lambda_D \mathcal{E}_D \ .$$

Since each particle has three degrees of freedom, $\mathcal{E}_T$ is a function of $3n$ state variables. In this work, we use conjugate gradient to minimize it and dynamically compute λ_D so that the two energy terms have comparable influences [Fua and Leclerc, 1994b].

In general, $\mathcal{E}_{St}$, the image-correlation term, is not convex and conjugate gradient may not find a global minimum. The presence of $\mathcal{E}_D$, the deformation energy, which tends to convexify the energy landscape alleviates the problem but may prove insufficient for large sets of particles. In such cases, an effective way to achieve a desirable minimum of the objective function is to cluster the particles into smaller subsets, to optimize each subset independently, and to reiterate the process until a stable solution is found. In practice, this can be achieved either by using the metric of Section 3.3 to break the set of particles into smaller subsets or by histograming the median gray levels of the particles, as defined in Section 3.4.1, and grouping those that fall into the same histogram peaks. The latter makes

sense in the absence of surface markings because particles that have similar gray-levels are more likely to belong to the same underlying surfaces than particles that do not. This heuristic has been used to produce the results of Section 4 but is clearly too simple and will need refinement in future work. Note, however, that this grouping scheme is only used as a device to speed up the optimization: it need not be perfect since no final decision is based on it.

4 Results

In this section, we demonstrate the applicability of our method on a variety of imagery.

4.1 Evaluating Components of the Approach

We first use four stereo pairs of images of a head to illustrate the effectiveness of the initialization and clustering methods of Sections 3.2 and 3.3, given relatively clean stereo data. The 512x512 images, shown in Figure 8, are part of a sequence of forty that were acquired with a video camera over a period of a few seconds by turning around the subject who was trying to stand still. Camera models were later computed using standard photogrammetric techniques at the Institute for Geodesy and Photogrammetry, ETH-Zürich.

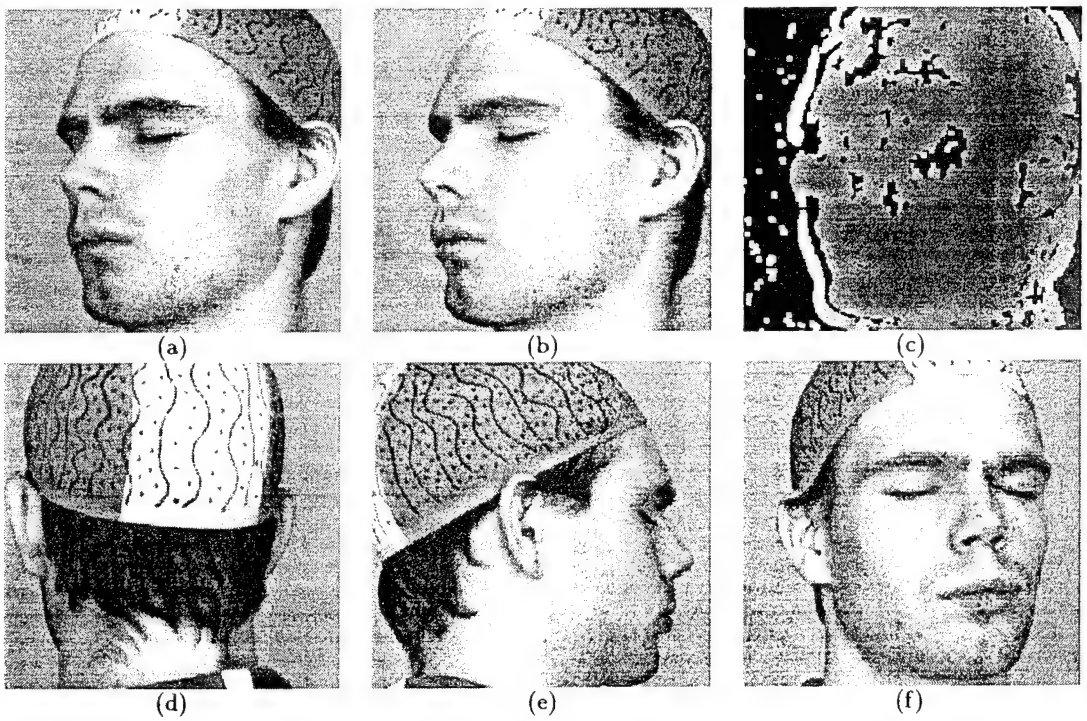


Figure 8: Head images (Courtesy of ETH-Zürich). (a,b) A stereo pair of a person's head as seen from one side. (c) The corresponding disparity map. Black indicates that no disparity value was computed, and lighter areas are further away than darker ones. Note that the disparities around the occluding contour on both the left and right sides of the head are erroneous. (d,e,f) The left images of three other stereo pairs of the same person taken from different viewpoints.

We ran our correlation-based algorithm [Fua, 1993]—once for each consecutive pair of images in the sequence—stored the resulting 3–D points in a 80x80x80 set of voxels and instantiated particles in all voxels containing at least 200 points. The results are shown in the first row of Figure 9. Because we use a large number of images, the main features of the head—including the nose, mouth, chin, ears and even the boundary of the skullcap—are

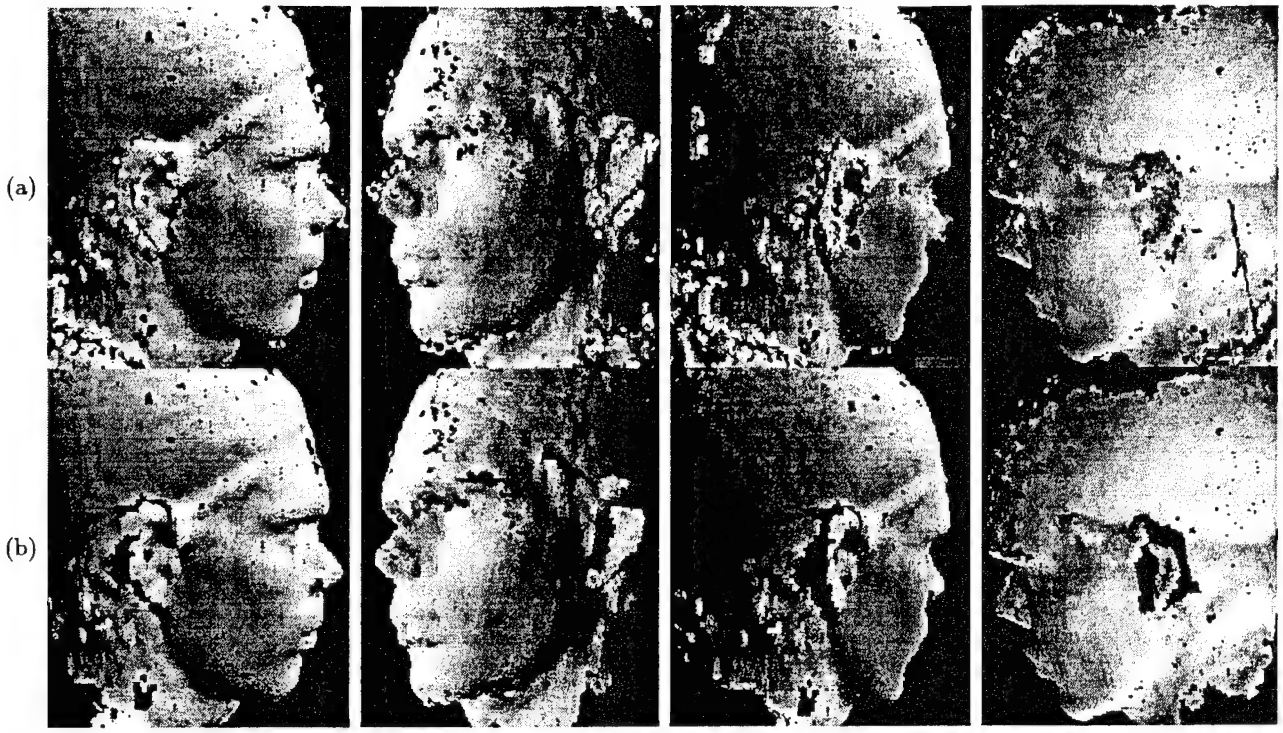


Figure 9: Modeling the head. Row (a): Four shaded views of the particles instantiated by fitting local surfaces to the 3–D points derived by correlating the images of Figure 8. Row (b): Similar views of the subset of particles that belong to the same global surface. Erroneous ones on the side of the nose and the back of the head have been eliminated.

clearly captured by our representation. However, because the correlation-based algorithm produced erroneous, but not random, disparities around occlusion boundaries, we also find a number of spurious particles around the nose, chin and back of the head. To get rid of them we computed the distance of Section 3.3 and eliminated all particles not having at least four neighbors within 1.2 times the median distance between neighbors, as shown in the second row of Figure 9. In this specific example, optimizing the positions of the particles yields a result that is virtually indistinguishable from the one presented here.

To demonstrate the accuracy that can be obtained by minimizing the total energy of Section 3.4, we use the 512x256 aerial images of Figure 10 and evaluate our results against the "ground truth" supplied to us by a photogrammetrist from Ohio State University. We ran our correlation-based algorithm, instantiated a set of particles using a 50x50x1 set of voxels, and refined their positions by minimizing the total energy of Equation 5. In Figure 10(h), we plot the vertical distance of the particles'centers from the triangulated surface defined by triangulating the control points. The Root Mean Square distance is 0.19 meter, which corresponds to an error in measured disparity that is smaller than half a pixel. Given the fact that the control points are not necessarily perfect themselves, this is the kind of performance one would expect of a precise stereo system [Güelch, 1988]. As discussed previously, in this simple case where only two images are used, our approach is equivalent to a correlation-based approach with deformable windows.

4.2 Reconstructing Complex 3–D Scenes

We now turn to the scene of Figure 11 whose modeling requires multiple viewpoints and a full 3–D representation. These images were acquired by setting a wheel and a box on a turntable and using the INRIA trinocular stereo rig to take seven triplets of images by rotating the turntable. In this case and in all views, the wheel presents surfaces

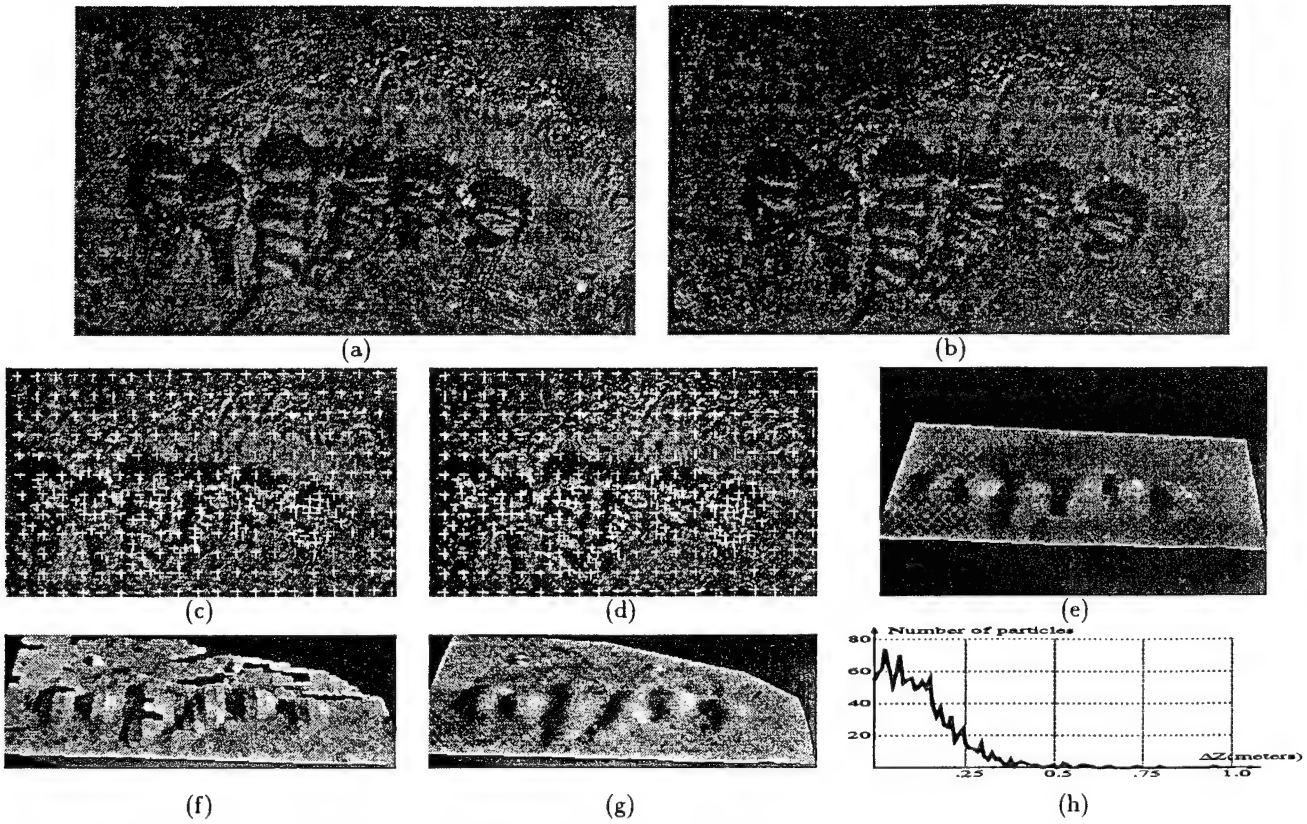


Figure 10: A stereo data set (courtesy of Ohio State University). (a,b) An aerial stereo pair. (c,d) Matched pairs of points hand-entered by a photogrammetrist. (e) Shaded view of the triangulated surface formed by the corresponding 3–D points. (f) The particles instantiated using a correlation-based disparity map after minimization of the total energy of the set. (g) Shaded view of the surface generated by triangulating the centers of the particles. (h) A plot of the vertical distance, in meters, from these centers to the "ground truth" surface of Figure 10(e). The RMS distance is equal to 0.2 meter, which corresponds to an error of approximately 0.4 pixel in disparity.

that are sharply slanted away from the cameras. The correlation data and the corresponding 3–D points, depicted by Figure 12(a), are much noisier than before. As a result, as shown in Figure 12(b), many more spurious particles are generated so that it becomes very difficult to distinguish the wheel from the base it rests on and from the erroneous fits. More specifically, there is no setting of the distance threshold that can cleanly separate the particles that sit on the wheel's surface from other ones. However, the optimization of the total energy of the set of particles produced enough of an improvement, shown in Figure 12(c), so that the clustering technique of Section 3.3 became able to effectively select the legitimate particles used to produce the shaded views of the figure's second row. This example demonstrates the ability of our object-centered representation to effectively pool the information from very different views to refine the representation beyond what could be done using only conventional stereo followed by robust surface fitting without reference to the original image data.

In Figure 13, we use the same setup and the same wheel as for the images of Figure 11, but we have added objects and use twelve triplets that span to a full 360 degrees of rotation of the turntable. The second row depicts the 3–D points computed by our simple correlation-based stereo algorithm. Note all the spurious points "floating" above the actual objects. For comparison's sake, we have verified that another correlation algorithm [Hannah, 1988], which is more sophisticated and is considered as one of the best ones currently available [Güelch, 1988], yields similar results. In fact, no setting of its parameters can effectively eliminate these points without also eliminating many of the real

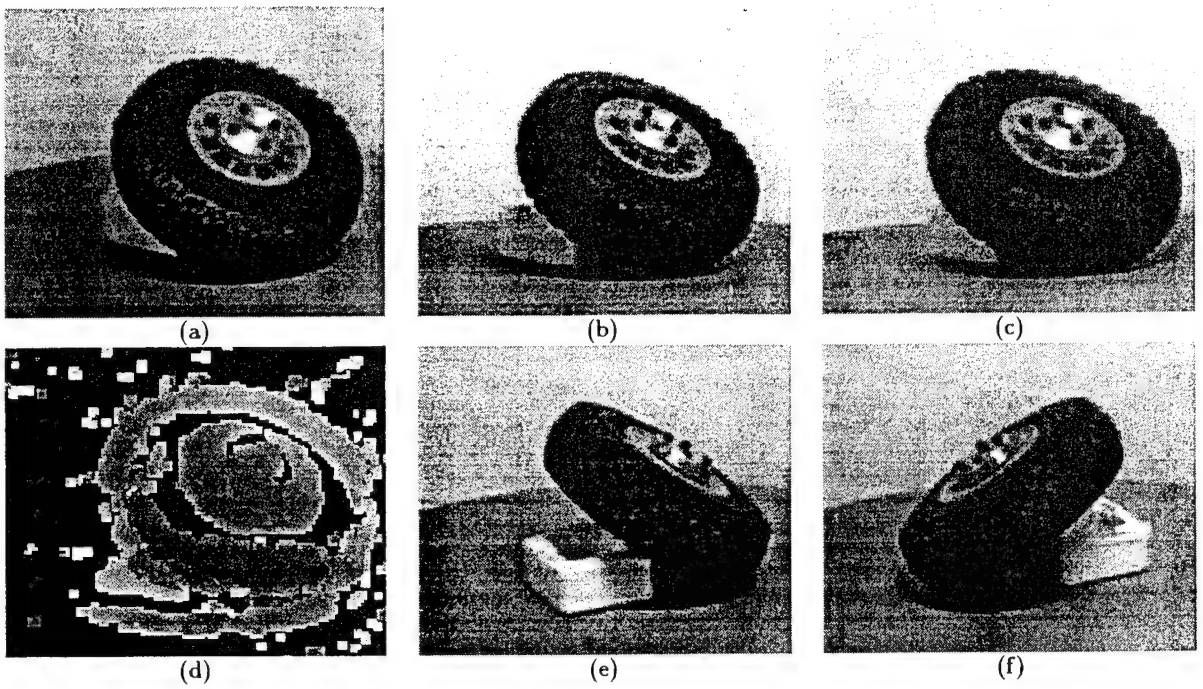


Figure 11: Images of wheel (Courtesy of INRIA). (a,b,c) A triplet of images of a wheel resting on a box sitting on a turntable. (c) Corresponding disparity map. Because the surface of the tire in the top left part of the wheel is slanted away from the camera, the disparities are of very poor quality in that part of the image. (e,f) First images of two other triplets acquired by rotating the turntable. The five other triplets used in this example were acquired for turntable positions that were intermediate between those of (e) and (f).

structures as long as one uses only pairs of images.

By running our system using the same parameters as before, except for the distance threshold, we generate the shaded representation shown in the third row of Figure 13. Note that the various objects—the wheel, the model of a brain, and the sides of the box on which they rest—appear clearly. However, our simple clustering mechanism does not pull them out as separate objects because it essentially uses the same threshold on surface curvature for the whole scene. In this case different thresholds are required for the different objects, and a more sophisticated heuristic—such as computing the distance threshold on a more local basis—would be required to achieve a complete segmentation. Finding optimal groups of particles can be recast as the problem of finding the best description of the scene in terms of this specific vocabulary given the input data. A possible approach would then be to use the Mininum Description Length [Rissanen, 1987] as was done by Leonardis *et al.* to extract surface patches from range data [1990]. Note also the hole in the wall of the wheel's tire in Figure 13(i). Careful examination of the original correlation data, reveals that 3–D data was particularly sparse there, presumably because that part of the tire is almost completely black. Such holes could potentially be filled by introducing a particle creation mechanism [Szeliski and Tonnesen, 1992].

In our final example, shown in Figure 14, we reconstruct a ground-level scene using three triplets of images acquired by the INRIA mobile robot. The ego-motion of the robot was computed by extracting 3–D segments and matching them across views [Zhang and Faugeras, 1990]. Note that the five main rocks in the scene, including one that overhangs, appear in the reconstruction.

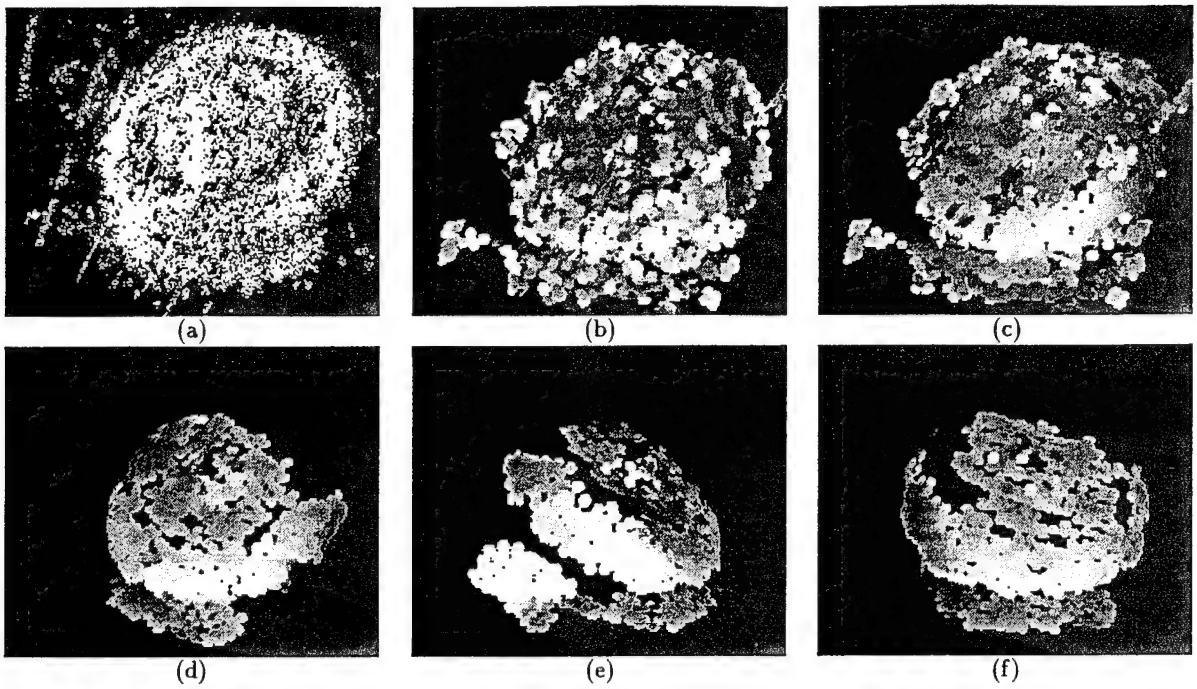


Figure 12: Modeling the wheel. (a) A subset of all the 3–D points computed by running a correlation-based algorithm on each of the image triplets. (b) The particles instantiated by fitting local surfaces to the 3–D points originating from seven noisy disparity maps. (c) The same particles after minimization of the total energy of the set. (e,f,g) Shading views of the set of legitimate particles after elimination of the spurious ones.

5 Conclusion

We have proposed a framework for 3–D surface reconstruction that can be used to model fully 3–D scenes from an arbitrary number of stereo views taken from vastly different viewpoints. By combining a particle-based representation, robust fitting, and optimization of an image-based objective function, we have been able to reconstruct surfaces without any a priori knowledge of their topology.

Our current implementation goes through three steps—initializing a set of particles from the input 3–D data, optimizing their location, and finally grouping them into global surfaces. We have demonstrated its competence and ability to merge information and thus to go beyond what can be done with conventional stereo alone.

However, as the quality of the input data decreases and the size of the problems we want to deal with increases, some of the current heuristics—specifically the ones used to cluster the particles for the purposes of both optimizing the set and rejecting outliers—may prove too simple. To push the method forward, it will be necessary to develop more sophisticated grouping techniques and to introduce new heuristics to fill in holes in the original correlation data. The basic framework, however, will remain because it provides the primitives required to implement these additional capabilities.

Acknowledgments

Support for this research was provided by various contracts from the Advanced Research Projects Agency. We wish to thank Yvan Leclerc and Richard Szeliski for their ideas—particularly the suggestion to use hash-tables to deal with neighborhood relationships—and active support in developing the approach reported here. We also wish to

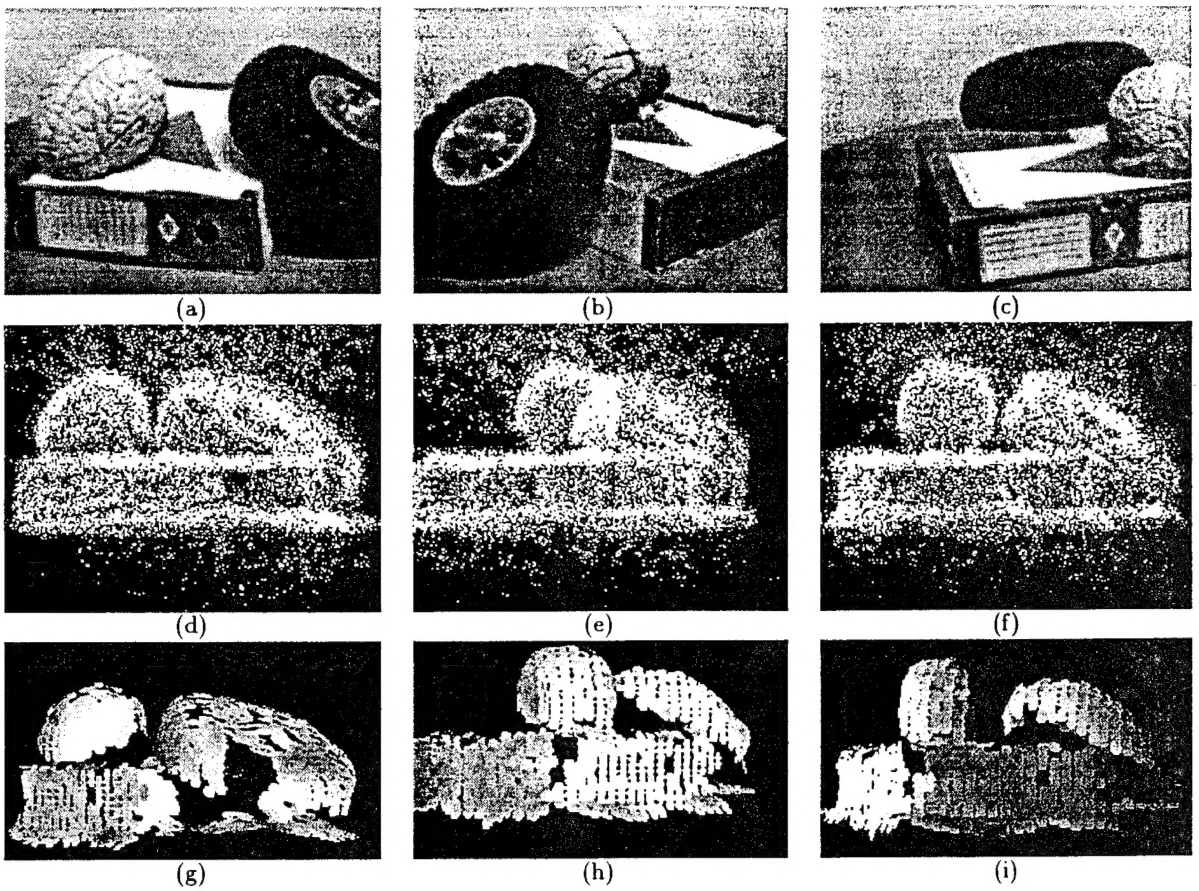


Figure 13: Images of a complex indoor scene (Courtesy of INRIA). (a,b,c) First images of three out of twelve image triplets. The three viewpoints are 120 degrees apart. (d,e,f) A subset of all the 3–D points computed by running a correlation-based algorithm on each of the image triplets. (g,h,i) Shaded views of the particle set after refinement and clustering.

thank Olivier Faugeras and the members of the INRIA Robotvis group, as well as Armin Gruën and the members of his group, who have supplied us with most of the images and calibration data that are presented here and that have proved extremely valuable to our research effort.

References

[Beaton and Turkey, 1974] A. E. Beaton and J.W. Turkey. The Fitting of Power Series, meaning Polynomials, Illustrated on Band-Spectroscopic Data. *Technometrics*, 16:147–185, 1974.

[Black and Rangarajan, 1994] M. J. Black and A. Rangarajan. The Outlier Process: Unifying Line Processes and Robust Statistics. In *Conference on Computer Vision and Pattern Recognition*, pages 121–128, Seattle, WA, June 1994.

[Chen and Medioni, 1994] Y. Chen and G. Medioni. Surface Descriptions of Complex Objects from Multiple Range Images. In *Conference on Computer Vision and Pattern Recognition*, pages 437–442, Seattle, WA, June 1994.

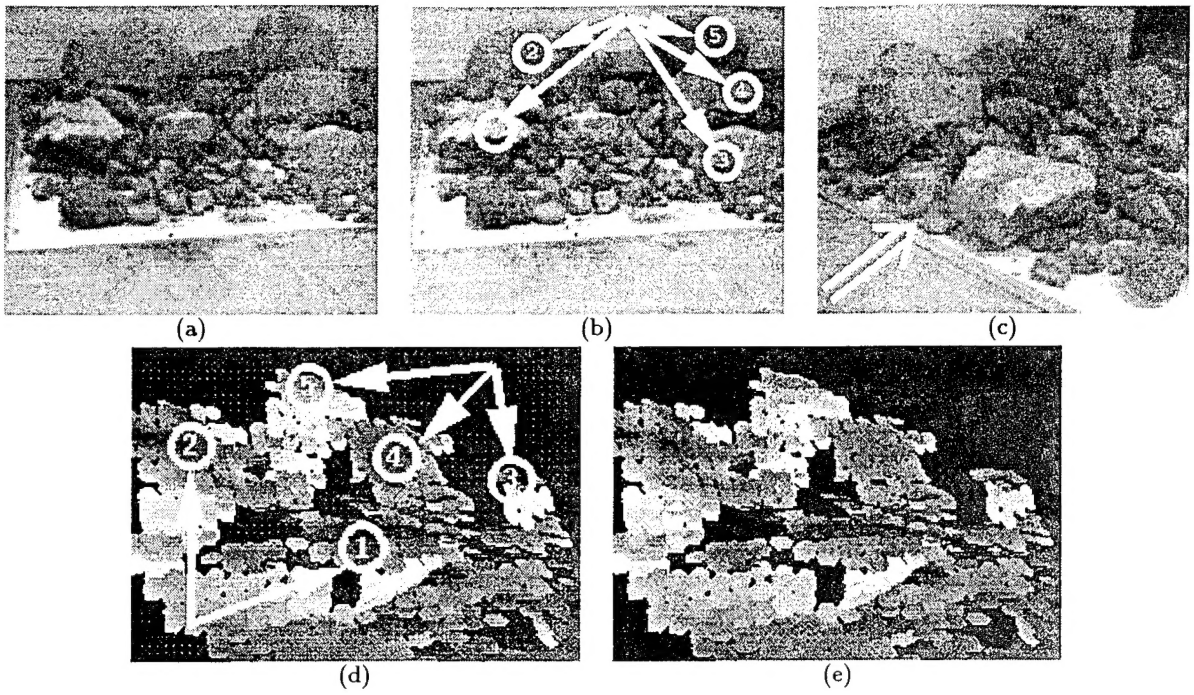


Figure 14: Modeling a pile of rocks. (a) The first image of a triplet (Courtesy of INRIA). (b) The same image with the five largest rocks labeled from 1 to 5. (c) The first image of a triplet taken from a different viewpoint. (e) Shaded view of the particle set after refinement and clustering seen from a viewpoint located on the left side of the rock pile, as indicated by the double white arrow in (c). (f) The same view with the five largest rocks labeled as in (b). Note that the overhang of rock number 1 is well recovered, a result that would be difficult to achieve using a 2-1/2–D representation.

[Cohen *et al.*, 1991] I. Cohen, L. D. Cohen, and N. Ayache. Introducing New Deformable Surfaces to Segment 3d Images. In *Conference on Computer Vision and Pattern Recognition*, pages 738–739, 1991.

[Delingette *et al.*, 1991] H. Delingette, M. Hebert, and K. Ikeuchi. Shape Representation and Image Segmentation using Deformable Surfaces. In *Conference on Computer Vision and Pattern Recognition*, pages 467–472, 1991.

[Devernay and Faugeras, 1994] F. Devernay and O. D. Faugeras. Computing Differential Properties of 3–D Shapes from Stereoscopic Images without 3–D Models. In *Conference on Computer Vision and Pattern Recognition*, pages 208–213, Seattle, WA, June 1994.

[Ferrie *et al.*, 1992] F. P. Ferrie, J. Lagarde, and P. Whaite. Recovery of Volumetric Object Descriptions from Laser Rangefinder Images. In *European Conference on Computer Vision*, Genoa, Italy, April 1992.

[Fua and Leclerc, 1994a] P. Fua and Y. G. Leclerc. Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading. *International Journal of Computer Vision*, 1994. In press, available as Tech Note 535, Artificial Intelligence Center, SRI International.

[Fua and Leclerc, 1994b] P. Fua and Y. G. Leclerc. Using 3–Dimensional Meshes To Combine Image-Based and Geometry-Based Constraints. In *European Conference on Computer Vision*, pages 281–291, Stockholm, Sweden, May 1994.

[Fua, 1993] P. Fua. A Parallel Stereo Algorithm that Produces Dense Depth Maps and Preserves Image Features. *Machine Vision and Applications*, 6(1):35–49, Winter 1993.

[Güelch, 1988] E. Güelch. Results of Test on Image Matching of ISPRS WG III / 4. *International Archives of Photogrammetry and Remote Sensing*, 27(III):254–271, 1988.

[Hannah, 1988] M.J. Hannah. Digital Stereo Image Matching Techniques. *International Archives of Photogrammetry and Remote Sensing*, 27(III):280–293, 1988.

[Hoff and Ahuja, 1987] W. Hoff and N. Ahuja. Extracting Surfaces from Stereo Images. In *International Conference on Computer Vision*, June 1987.

[Hoppe *et al.*, 1992] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface Reconstruction from Unorganized Points. In *Computer Graphics (SIGGRAPH)*, volume 26, pages 71–78, July 1992.

[Hotz *et al.*, 1993] B. Hotz, Z. Zhang, and P. Fua. Incremental construction of local D.E.M for an autonomous planetary rover. In *Workshop on Computer Vision for Space Applications*, Antibes, France, September 1993.

[Koh *et al.*, 1994] E. Koh, D. Metaxas, and N. Badler. Hierarchical Shape Representation Using Locally Adaptative Finite Elements. In *European Conference on Computer Vision*, Stockholm, Sweden, May 1994.

[Leonardis *et al.*, 1990] A. Leonardis, A. Gupta, and R. Bajcsy. Segmentation as the Search for the Best Description of the Image in Terms of Primitives. In *International Conference on Computer Vision*, pages 121–125, Osaka, Japan, December 1990.

[Lowe, 1991] D. G. Lowe. Fitting Parameterized Three-Dimensional Models to Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(441-450), 1991.

[McInerney and Terzopoulos, 1993] T. McInerney and D. Terzopoulos. A Finite Element Model for 3D Shape Reconstruction and Nonrigid Motion Tracking. In *International Conference on Computer Vision*, pages 518–523, Berlin, Germany, 1993.

[Nishihara, 1984] H.K. Nishihara. Practical Real-Time Imaging Stereo Matcher. *Optical Engineering*, 23(5), 1984.

[Park *et al.*, 1994] J. Park, D. Metaxas, and A. Young. Deformable Models with Parameter Functions: Application to Heart-Wall Modeling. In *Conference on Computer Vision and Pattern Recognition*, pages 437--442, Seattle, WA, June 1994.

[Pentland and Sclaroff, 1991] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:715–729, 1991.

[Pentland, 1990] A. Pentland. Automatic Extraction of Deformable Part Models. *International Journal of Computer Vision*, 4(2):107–126, March 1990.

[Rissanen, 1987] J. Rissanen. *Encyclopedia of Statistical Sciences*, volume 5, chapter Minimum-Description-Length Principle, pages 523–527. John Wiley and Sons, New York, New York, 1987.

[Robert *et al.*, 1992] L. Robert, R. Deriche, and O.D. Faugeras. Dense Depth Recovery From Stereo Images. In *European Conference on Artificial Intelligence*, pages 821–823, Vienna, Austria, August 1992.

[Sander and Zucker, 1990] Peter T. Sander and Steven W. Zucker. Inferring Surface Trace and Differential Structure from 3-D Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(9):833–854, September 1990.

[Stewart, 1994] C. V. Stewart. A New Robust Operator for Computer Vision: Application to Range Data. In *Conference on Computer Vision and Pattern Recognition*, pages 167–173, Seattle, WA, June 1994.

[Stokely and Wu, 1992] E. M. Stokely and S. Y. Wu. Surface parameterization and curvature measurement of arbitrary 3-D objects: five practical methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):833–839, August 1992.

[Szeliski and Tonnesen, 1992] R. Szeliski and D. Tonnesen. Surface Modeling with Oriented Particle Systems. In *Computer Graphics (SIGGRAPH)*, volume 26, pages 185–194, July 1992.

[Terzopoulos and Metaxas, 1991] D. Terzopoulos and D. Metaxas. Dynamic 3D models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(703-714), 1991.

[Terzopoulos and Vasilescu, 1991] D. Terzopoulos and M. Vasilescu. Sampling and reconstruction with adaptive meshes. In *Conference on Computer Vision and Pattern Recognition*, pages 70–75, 1991.

[Vemuri and Malladi, 1991] B. C. Vemuri and R. Malladi. Deformable models: Canonical parameters for surface representation and multiple view integration. In *Conference on Computer Vision and Pattern Recognition*, pages 724–725, 1991.

[Zhang and Faugeras, 1990] Z. Zhang and O.D. Faugeras. Tracking and Motion Estimation in a Sequence of Stereo Frames. In L.C. Aiello, editor, *European Conference on Artificial Intelligence*, pages 747–752, Stockholm, Sweden, August 1990.